

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: Робототехнічна підсистема кліматконтролю розумного дому

Виконав: студент другого курсу, групи ІК-81мп
(шифр групи)

Гайовий Олександр Анатолійович
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, д.т.н., ст.н.с., Борукаєв З.Х.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант НК к.т.н., доцент, Пасько В.П.
(назва розділу) (науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент доцент, к.т.н., доцент Ліщук К.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ І.Р. Пархомей
(підпис)

«__» _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Гайовому Олександру Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Робототехнічна підсистема кліматконтролю розумного
дому

науковий керівник дисертації Борукаєв Зелимхан Харитонович, д.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28» 10 2019 р. № 3770-с

2. Термін подання студентом дисертації 18.11.2019

3. Об'єкт дослідження Робототехнічна підсистема

4. Предмет дослідження Методи контролю клімату розумного дому

5. Перелік завдань, які потрібно розробити 1. Аналіз існуючих підсистем

2. Аналіз робототехнічних систем

3. Аналіз систем кліматконтролю розумного дому

4. Створення робототехнічної підсистеми кліматконтролю

5. Управління кліматконтролем

6. Орієнтовний перелік ілюстративного матеріалу 1. Схема робототехнічної підсистеми; 2. Схема управління кліматконтролем; 3. Схема інформаційних потоків; 4. Скріншоти роботи системи.

7. Орієнтовний перелік публікацій Гайовий О.А., Nauka bez granic, Робототехнічна система “Розумного дому” на базі Arduino

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
НК	Пасько В.П., доцент		
Перевірка на співпадіння	Лісовиченко О.І., доцент		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Формування проблематики	02.09.2019 – 08.09.2019	
2	Аналіз проблематики	09.09.2019 – 15.09.2019	
3	Постановка задачі	16.09.2019 – 22.09.2019	
4	Створення логіки управління кліматконтролем	23.09.2019 – 29.09.2019	
5	Створення структури робототехнічної системи	30.09.2019 – 06.10.2019	
6	Розробка ПЗ	07.10.2019 – 13.10.2019	
7	Тестування та покращення ПЗ	14.10.2019 – 20.10.2019	
8	Практичне застосування ПЗ	21.10.2019 – 27.10.2019	

Студент

(підпис)

Гайовий О.А.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Борукаєв З.Х.

(ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему в області автоматизації системи кліматконтролю в Розумному домі, показано основні особливості існуючих рішень та додатків, їх переваги та недоліки.

Існуючі рішення, представлені на ринку, мають зовелику ціну, або невеликий спектр функцій. Вони можуть виконати певний спектр функцій, але недостатній для комплексної роботи з кліматом.

Визначено завдання для робототехнічної системи кліматконтролю, відібраний мікроконтролер та спектр виконуємих функцій, які найбільш підходять для даної задачі. Описано структуру мікроконтроллера та проведено тестування програми.

Ця система надає вам автоматизоване керування кліматом у вашому домі. Дозволяє керувати кліматом з будь-якого пристрою.

Ключові слова: мікроконтролер, Arduino, схема, датчик.

Розмір пояснювальної записки – 92 аркушів, містить 68 ілюстрацій, 22 таблиць, 4 додатка.

ABSTRACT

The paper considers the problem in the field of climate control system automation in the Smart House, shows the main features of existing solutions and applications, their advantages and disadvantages.

Existing solutions on the market have too high a cost or a small range of features. They can perform a range of functions, but not enough to handle the climate.

The tasks for the robotic climate control system, the selected microcontroller and the range of functions that are most suitable for this task are determined. The structure of the microcontroller is described and the program tested.

This system provides you with automated climate control in your home. Allows you to control the climate from any device.

Keywords: microcontroller, Arduino, circuit, sensor.

Explanatory note size – 92 pages, contains 68 illustrations, 22 tables, 4 applications.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ.....	6
1.1. Технологія “Розумний дім”.....	6
1.2. Об’єкт та предмет дослідження.....	7
1.3. Сервіси розумного дому.....	7
1.4. Огляд існуючих рішень.....	18
1.5. Підсистема кліматконтролю на базі Arduino.....	20
ВИСНОВКИ ДО РОЗДІЛУ	21
РОЗДІЛ 2. КОМПОНЕНТИ ARDUINO UNO TA ADRUINO IDE.....	22
2.1. Ознайомлення з Arduino Uno.....	22
2.2. Компоненти Arduino Uno	22
2.3. Конфігурація пінів.....	26
2.4. Схема Arduino Uno	27
2.5. Модулі та інші деталі Arduino.....	31
2.6. Arduino IDE.....	43
ВИСНОВКИ ДО РОЗДІЛУ	60
РОЗДІЛ 3. ПРАКТИЧНЕ ВИКОРИСТАННЯ ПІДСИСТЕМИ КЛІМАТКОНТРОЛЮ.....	61
3.1. Реалізація та підключення до мікроконтролера.....	61
3.2. Web додаток.....	62
ВИСНОВКИ ДО РОЗДІЛУ	69
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ.....	70
4.1.Опис ідеї проекту.....	72
4.2. Технологічний аудит проекту.....	74
4.3. Розробка ринкової стратегії.....	84

ВИСНОВКИ ДО РОЗДІЛУ	88
ВИСНОВКИ.....	89
ПЕРЕЛІК ПОСИЛАНЬ.....	90

ВСТУП

Сьогодні, в епоху сучасних технологій, важливість автоматизації та комп'ютеризації в усіх сферах набуває все більшого значення. Будинок, оснащений найсучаснішими автоматизованими системами, повинен спростити наше життя, а не ускладнювати його. У більшості випадків метою вдосконалення технологій є захист людей та забезпечення їм необхідного комфорту. Тому інтеграція інтелектуального обладнання в домашню інфраструктуру просто спрощує наше життя та заощаджує наш час та гроші опосередковано.

Розумний будинок оснащений безліччю систем управління електроенергією, витратами на обслуговування будинку, електронними платіжними системами, різними режимами моніторингу тощо. Автоматизація робить будинок практичним та простим у використанні багатофункціональним комплексом, гармонійно та цілісно інтегруючи всю домашню аудіо та відеосистему, комунікації, кондиціонування, освітлення, безпеку, Інтернет тощо. Через це, виникає дуже сильний попит на даного роду пристрої.

Протягом цієї дисертації, будуть розглянуті та проаналізовані існуючі аналоги підсистем кліматконтролю розумного дому, освітлюється процес створення та розробки робототехнічної підсистеми кліматконтролю на базі мікроконтролера Arduino UNO та розглядається процес управління кліматом, за допомогою створеного WEB сервісу, що контролює мікроконтролер.

ПЕРЕЛІК СКОРОЧЕНЬ

HTML – HyperText Markup Language

API – Application programming interface

ПЗ – Програмне забезпечення

ІОТ – Internet of things

ОС – Операційна система

ПК – Персональний комп'ютер

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1. Технологія “Розумний дім”

Розумний дім є житловим розширенням автоматизації будівель і включає контроль та автоматизацію всіх вбудованих технологій. Він визначає місце проживання, яке має прилади, освітлення, опалення, кондиціонування, телевізори, комп’ютери, розважальні системи, великі побутові прилади, такі як пральні машинки та холодильники / морозильні камери, охоронні та камерні системи, здатні спілкуватися один з одним та контролюватися ними віддалено за графіком, телефоном, мобільним телефоном або Інтернетом. Ці системи складаються з комутаторів і датчиків, підключених до центрального концентратора, керованого мешканцем будинку, використовуючи настінний термінал або мобільний блок, підключений до хмарних сервісів Інтернету.

Встановлення розумних продуктів забезпечує зручність та економію часу, грошей та енергії. Такі системи є пристосувальними та регульованими для задоволення постійних мінливих потреб мешканців будинку. У більшості випадків її інфраструктура є достатньо гнучкою для інтеграції з широким спектром пристроїв різних постачальників та стандартів. Основна архітектура дозволяє вимірювати домашні умови, обробляти впорядковані дані, використовуючи датчики, що підтримуються мікроконтролером, для вимірювання домашніх умов та приводів для моніторингу вбудованих пристроїв для дому. Популярність та проникнення концепції розумного дому зростає з хорошими темпами, оскільки вона стала частиною модернізації та зменшення тенденцій витрат. Це досягається за рахунок вбудовування можливості підтримувати централізований журнал подій, виконувати процеси машинного навчання для забезпечення основних елементів витрат, економії рекомендацій та інших корисних звітів.

1.2. Об'єкт та предмет дослідження

Об'єктом виступає робототехнічна підсистема, а предметом – методи контролю клімату розумного дому.

1.3. Сервіси Розумного дому

Вимірювання домашніх умов

Типовий розумний будинок оснащений набором датчиків для вимірювання домашніх умов, таких як: температура, вологість, світло тощо. Кожен датчик призначений для зйомки одного або декількох вимірювань. Температуру і вологість можна виміряти одним датчиком, інші датчики обчислюють коефіцієнт світла для даної ділянки та відстань від нього до кожного предмета, що піддається впливу. Усі датчики дозволяють зберігати дані та візуалізувати їх, щоб користувач міг їх переглядати в будь-якому місці та в будь-який час. Для цього він включає процесор сигналу, інтерфейс зв'язку та хост на хмарній інфраструктурі.

Управління побутовою технікою

У розумному домі, часто є хмарний сервіс для управління побутовою технікою, який розміщуватиметься у хмарній інфраструктурі. Служба керування дозволяє користувачеві керувати виходами розумних приводів, пов'язаних з побутовою технікою, наприклад, лампами та вентиляторами. Розумні приводи – це пристрої, такі як клапани та вимикачі, які виконують такі дії, як включення чи вимкнення речей або налаштування робочої системи. Приводи забезпечують різноманітні функції, такі як обслуговування клапанів увімкнення / вимкнення, позиціонування до відкритого відсотка, модуляція для управління змінами в умовах потоку, аварійне відключення (ESD). Щоб активувати привід, приводу видається команда, яка контролює певний прилад.

Основні компоненти

Щоб включити всі описані вище дії та управління даними, система складається з наступних компонентів, як описано на рис. 1.1.

- а. Датчики для збору внутрішніх і зовнішніх домашніх даних та вимірювання домашніх умов. Ці датчики підключені до самого будинку та до приєднаних домашніх пристроїв. Дані датчиків збираються та постійно
- б. передається через локальну мережу, на сервер розумного дому.
- с. Процесори для виконання локальних та інтегрованих дій. Він також може бути підключений до хмари для додатків, що вимагають розширених ресурсів.
- д. в. Колекція програмних компонентів, завершених як API, дозволяючи зовнішнім програмам виконувати її, якщо вона відповідає заданому формату параметрів. Такий API може обробляти дані датчиків або керувати необхідними діями.

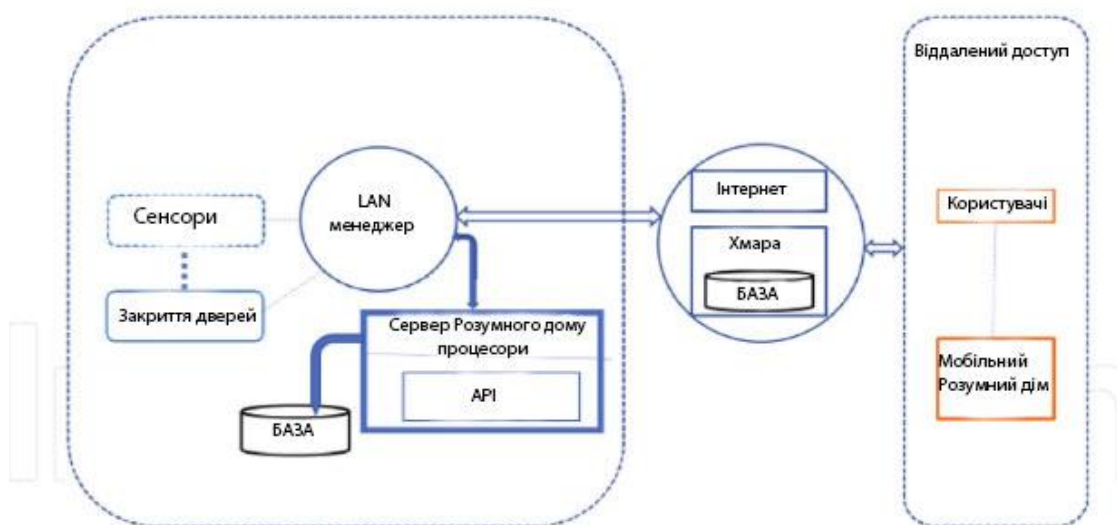


Рисунок 1.1. Класична схема Розумного дому

- е. Приводи забезпечення та виконання команд на сервері чи інших пристроях управління. Він переводить необхідну активність у синтаксис команд,

які пристрій може виконати. Під час обробки даних отриманих датчиків завдання перевіряє, чи якесь правило стало істинним. У такому випадку система може запустити команду на належний процесор пристрою.

f. База даних для зберігання оброблених даних, зібраних з датчиків та хмарних сервісів. Він також буде використовуватися для аналізу даних, представлення даних та візуалізації. Оброблені дані зберігаються у доданій базі даних для подальшого використання

Огляд Інтернету речей (IoT)

Парадигма Інтернет речей (IoT) відноситься до пристроїв, підключених до мережі. Пристрої – це об'єкти, такі як датчики та пускачі, оснащені телекомунікаційним інтерфейсом, процесорним блоком, обмеженим сховищем та програмними програмами. Це дозволяє інтегрувати об'єкти в Інтернет, встановлюючи взаємодію між людьми та пристроями між пристроями. Ключова технологія IoT включає радіочастотну ідентифікацію (RFID), сенсорну технологію та інтелектуальну технологію. RFID є основою та мережевим ядром побудови IoT. Її можливості обробки та зв'язку разом з унікальними алгоритмами дозволяють інтегрувати різноманітні елементи для роботи як інтегрованого підрозділу, але в той же час дозволяють легко додавати та видаляти компоненти за допомогою мінімальний вплив, роблячи IoT надійним, але гнучким, щоб поглинати зміни в оточенні та вподобаннях користувачів. Для мінімізації використання пропускної здатності використовується легка версія JSON для внутрішніх компонентів та зовнішніх повідомлень.

Хмарні обчислення та їх внесок у IoT та розумний дім

Хмарні обчислення – це спільний обсяг обчислювальних ресурсів, готових надавати різноманітні обчислювальні послуги на різних рівнях, від базової інфраструктури до найскладніших прикладних сервісів, легко розподіляються та звільняються з мінімальними зусиллями або взаємодією постачальника послуг. На практиці він управляє ресурсами обчислювальної техніки, зберігання та зв'язку, якими ділиться декілька користувачів у віртуальному та ізольованому середовищі. На рис. 1.2. зображена загальна парадигма хмари.



Рисунок 1.2. Парадигма хмарних обчислень

IoT та розумний дім можуть скористатися широкими ресурсами та функціоналами хмари, щоб компенсувати його обмеження в зберіганні, обробці, спілкуванні, підтримці у виборі попиту, резервного копіювання та відновлення. Наприклад, хмара може підтримувати управління послугами IoT та виконувати та виконувати додаткові додатки, використовуючи дані, створені нею. Розумний дім можна стиснути та зосередити увагу лише на основних та найважливіших

функціях, і таким чином мінімізувати місцеві домашні ресурси та покладатися на хмарні можливості та ресурси. Розумний дім та IoT будуть зосереджені на зборі даних, базовій обробці та передачі в хмару для подальшої обробки. Щоб вирішити проблеми з безпекою, хмара може бути приватною для високо захищених даних і загальнодоступною для решти. Але, швидше, баланс між локальними та центральними обчисленнями разом з оптимізацією споживання ресурсів. Обчислювальне завдання може бути виконано на пристроях IoT та розумного дому або передано в хмару. Де проводити обчислення, залежить від накладних компромісів, доступності даних, залежності даних, кількості транспортування даних, залежності зв'язку та безпеки. З одного боку, потрібна обчислювальна модель, що включає хмару, IoT та розумний дім, повинна мінімізувати всю вартість системи, зазвичай, приділяючи більше уваги скороченню споживання ресурсів вдома. З іншого боку, модель послуг IoT та розумних домашніх обчислень повинна покращити користувачів IoT, щоб задовольнити їх попит під час використання хмарних додатків та вирішення складних проблем, що виникають внаслідок нової моделі IoT, розумного дому та хмарних служб. Деякі приклади медичних послуг, що надаються інтеграція хмари та IoT: належне управління інформацією, обмін електронними записами про охорону здоров'я дозволяють забезпечити високоякісні медичні послуги, керувати даними датчиків охорони здоров'я, робить мобільні пристрої придатними для доставки даних про охорону здоров'я, безпеки, конфіденційності та надійності, підвищуючи безпеку медичних даних та доступність послуг та надмірність послуг та надання допомоги в режимі реального часу та хмарне виконання мультимедійних медичних послуг.

Централізована обробка подій, система на основі правил

Розумний дім та IoT багаті сенсорами, які генерують масивні потоки даних у вигляді повідомлень чи подій. Обробка цих даних вище можливостей людини. Отже, системи обробки подій були розроблені та використовуються для швидшого реагування на класифіковані події.

Користувач може визначити правило, викликане подіями, і контролювати належну доставку послуг. Правило складається з умов події, структури подій та інформації, пов'язаної з кореляцією, яка може поєднуватися для моделювання складних ситуацій. Він був реалізований у типовому розумному будинку та довів свою придатність для сервісно орієнтованої системи. Система може обробляти велику кількість подій, виконувати функції для моніторингу, навігації та оптимізації процесів у режимі реального часу. Він виявляє та аналізує аномалії чи винятки та створює реактивні / активні реакції, такі як попередження та запобігання пошкодженню. Ситуації моделюються зручним для користувача інтерфейсом моделювання для ініційованих подіями правил. Якщо потрібно, вони розбивають їх на прості, недостатньо стійкі елементи. Запропонована модель може бути легко інтегрована в розподілену та орієнтовану на сервіс платформу обробки подій. Процес оцінювання ініціюється подіями, що доставляють останнє стан та інформацію з відповідного середовища. Результат – графік рішення, що представляє правило. Він може розбити складні ситуації на прості умови та поєднати їх між собою, склавши складні умови. Вихід – це подія відповіді, яка виникає, коли правило спрацьовує. Запущені події можуть бути використані як вхід для інших правил для подальшої оцінки. Шаблони подій виявляються, коли відбувається декілька подій і відповідають попередньо визначеному шаблону. Завдяки графічній моделі та модульному підходу до побудови правил правила можна легко адаптувати до змін домену. Нові умови події або шаблони подій можуть бути додані або вилучені з моделі правила.

Правила виконуються службами подій, які доповнюють механізм управління подіями та обробляють результат оцінки. Щоб забезпечити наявність відповідних ресурсів обробки, система може працювати в розподіленому режимі, на декількох машинах та полегшувати інтеграцію із зовнішніми системами. Визначення взаємозв'язків та залежностей між подіями, що мають значення для обробки правил, виконуються за допомогою наборів послідовностей, генерованих механізмом правил. Двигун правил будує послідовності подій, що відповідають конкретній умові правила, щоб дозволити асоціювати події за їх контекстними даними. Правила автоматично виконують дії у відповідь, коли зазначені умови дотримуються. Дії породжують події відповіді, які викликають активність реакції. Шаблони подій можуть відповідати тимчасовим послідовностям подій, що дозволяє описувати домашні ситуації, коли події є актуальними. Наприклад, коли двері тримаються відкритими занадто довго. У цій моделі відомі такі проблеми: структура оброблюваних подій та даних, конфігурація служб та адаптерів для етапів обробки, включаючи вхідні та вихідні параметри, інтерфейси для зовнішніх системи зондування даних та реагування на виконання транзакцій, структура оброблюваних подій та даних, перетворення даних, аналіз даних та постійність. Це дозволяє моделювати, які події повинні бути оброблені службою правил і як події відповіді повинні бути передані іншим службам подій. Процес простий: дані збираються та отримуються від адаптерів, які пересилають події до служб подій, які їх споживають. Спочатку події збагачуються, щоб підготувати дані події до обробки правил. Наприклад, події відповіді надсилаються службі для надсилання сповіщень агенту дзвінків або службам, які передають сповіщення про затримку подій та оновлення подій назад в систему управління подіями.

Зупинимось на інтеграції інтелектуального будинку, IoT та облачного складання для визначення нової обчислювальної парадигми. Розглянемо

синергію між цими трьома концепціями та шукаємо шляхи інтеграції їх у нову всеосяжну парадигму, використовуючи її загальні основні поняття, а також її унікальні атрибути, щоб дозволити виконання нових процесів, які не могли бути оброблені інакше. На рис 1.3. зображено вдосконалені основні компоненти розумного дому та їх взаємозв'язок. На лівому блоці, в середовищі розумного будинку, ми бачимо типові пристрої, підключені до локальної мережі [LAN]. Це дає змогу спілкуватися між пристроями та поза ними. Підключений до локальної мережі – це сервер та його база даних. Сервер контролює пристрої, реєструє його діяльність, надає звіти, відповідає на запити та виконує відповідні команди. Для більш всебічних або загальних завдань сервер розумного дому передає дані в хмару та віддалено активує завдання в ньому за допомогою API, процесів інтерфейсу програмування програм. Крім того, побутова техніка IoT підключена до Інтернету та до локальної мережі, і таким чином розширює розумний будинок, включаючи IoT. активувати завдання.

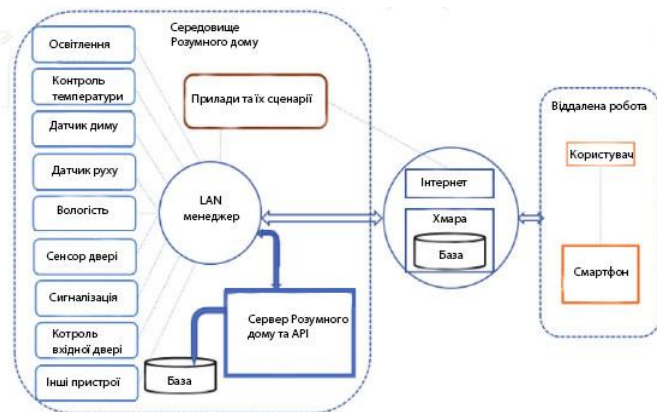


Рисунок 1.3. Розширений розумний дім – інтеграція розумного дому, IoT та хмарних обчислень

Щоб продемонструвати переваги розширеного розумного будинку, ми використовуємо RSA, надійний асиметричний алгоритм криптографії, який генерує відкритий та приватний ключ та шифрує / розшифровує повідомлення. За допомогою відкритого ключа кожен може зашифрувати повідомлення, але

лише ті, хто має приватний ключ, можуть розшифрувати надіслане повідомлення. Генерування ключів та шифрування / розшифрування повідомлень включає в себе обширні обчислення, що потребують значного простору пам'яті та потужності для обробки. Тому його зазвичай обробляють на потужних комп'ютерах, побудованих для впорядкування необхідних ресурсів. Однак, через обмежені ресурси, запускати RSA на пристрої IoT майже неможливо, і, таким чином, це відкриває розрив у безпеці в Інтернеті, де зловмисники можуть легко використовувати. Щоб впоратися з цим, ми поєднуємо в собі потужність локальних процесорів розумного дому для обчислення деяких обчислень RSA та передачі складніших обчислювальних завдань для обробки в хмарі. Потім результати будуть передані назад на IoT – датчик, який буде складено і зібрано разом, щоб генерувати код шифрування / дешифрування RSA і так закрити згаданий проміжок безпеки IoT. Цей приклад демонструє потік даних серед передових компонентів розумного дому. Де кожен компонент виконує свій власний стек операцій для отримання унікального результату. Однак у випадку складних і тривалих завдань воно розділить завдання на підзавдання, які виконуються більш потужними компонентами. Посилаючись на приклад RSA, пристрій IoT ініціює необхідність генерування ключа шифрування і, таким чином, надсилає повідомлення із запитом на додаток RSA, що працює в домашньому комп'ютері. Потім розумний домашній комп'ютер запитує програму «генерація простих чисел», що працює на хмарі, надати p і q прості числа. Як тільки p і q приймаються, формується код шифрування. На пізньому етапі пристрій IoT надсилає запит на розумний домашній комп'ютер для шифрування повідомлення, використовуючи недавно створений ключ шифрування RSA. Потім зашифроване повідомлення передається назад на пристрій IoT для подальшого виконання. Аналогічний сценарій може бути в протилежному напрямку, коли пристрій IoT отримує повідомлення, він може вимагати від

розумного будинку розшифрувати його. Для підсумовування, сценарії RSA зображують використання сили енергії хмарних обчислень, розумний захищений комп'ютер можливостей і в кінці обмеженої потужності пристрою IoT. Це доводить, що без цієї автоматичної співпраці RSA не змогла б бути виконана на рівні IoT. Більш практичним прикладом є те, коли кілька відокремлених приладів, такі як піч, повільна плита та каструля на вершині газової плити, працюють у виконанні прохання резидента. Мешканець отримує терміновий телефонний дзвінок і негайно йде з дому, не відключаючи активні прилади. Якщо відповідні IoT були налаштовані на автоматичне відключення на основі заздалегідь визначеного правила, це буде зроблено на рівні IoT. В іншому випадку розумний дім розуміє, що мешканець пішов з дому двері будинку були відчинені, а потім заблоковані, гараж відкритий, машина мешканця пішла, головна брама була відкрита, а потім закрита, нікого не було вдома і буде вимкнути всі активні пристрої, віднесені до ризику у разі його відсутності. Він надішле відповідне повідомлення до списку розсилки, визначеного для такого випадку.

Виявлення витоків води та її запобігання

Перший крок – розгортання датчиків води під кожним розумним потенційним джерелом витoku та автоматизованим датчиком водного клапана для всього будинку, що означає, що будинок розглядається як IoT. Якщо датчик води виявляє витік води, він посилає подію на концентратор, який запускає програму "відключити клапан". Потім додаток для управління будинком надсилає команду «вимкнути» всім пристроям IoT, визначеним як чутливі до зупинки води, а потім надсилає команду «вимкнути» на головний клапан води. Повідомлення для оновлення надсилається через систему обміну повідомленнями, які відображаються у списку сповіщень. Ця установка допомагає захищатись від сценаріїв, де джерело води знаходиться з водопроводу

будинку. Основна конфігурація передбачає інтеграцію за допомогою повідомлень та команд між розумним домом та системою керування IoT. Це демонструє залежність та отримані переваги від поєднання розумного будинку та IoT.

Датчики диму

У більшості будинків вже є типова колекція детекторів диму, але немає мосту для передачі даних від датчика в концентраційний будинок. Підключення цих датчиків до розумного домашнього додатка дає можливість комплексної системи виявлення диму. Він також розширений, щоб повідомити датчик елеватора, щоб заблокувати його використання через стан пожежі, і, таким чином, він ще більше поширюється на будь-який датчик IoT, який може бути активований через виявлене сповіщення про дим.

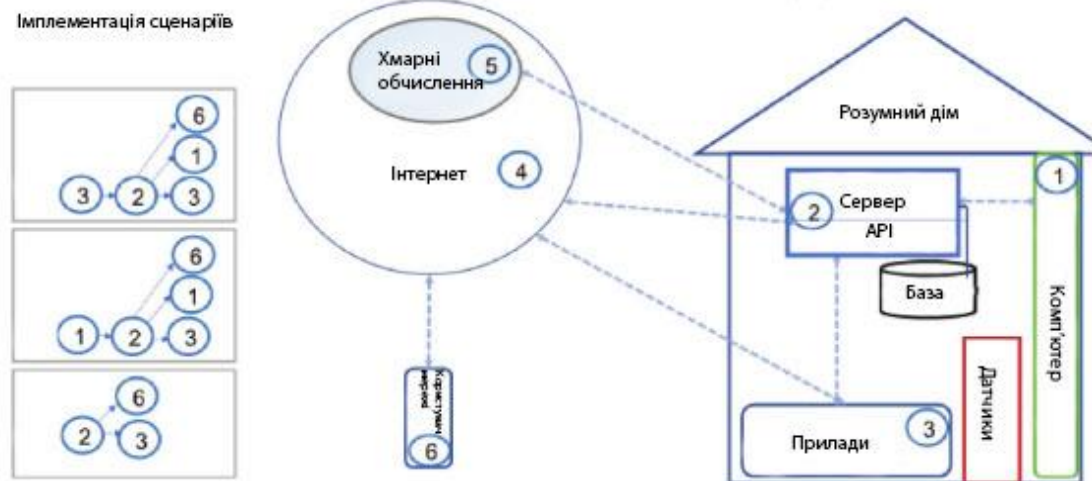


Рисунок 1.4. Розширена блок-схема реалізації розумного будинку.

Управління інцидентами для контролю побутової техніки

Розгляньте сценарій, коли ви виходите з дому, поки частина приладів все ще увімкнена. Якщо ваша відсутність буде досить довгою, частина приладів

може перегріватися і ось-ось вибухне. Щоб уникнути подібних ситуацій, ми підключаємо всі датчики пристроїв IoT до домашнього додатку, щоб, коли всі виїжджають з дому, автоматично регулювати всі датчики приладів, щоб уникнути пошкоджень. Зауважте, що індикація порожнього дому генерується програмою Smart Home, тоді як індикація "увімкнено" пристрою генерується IoT. Отже, такий сценарій можливий завдяки інтеграції між системами розумного дому та IoT.

1.4. Огляд існуючих рішень

Однією з найпопулярніших форм домашньої автоматизації є кліматичний контроль. Багато будинків зараз використовують автоматичні термостати та інші пристрої, щоб допомогти своїм будинкам комфортніше працювати. Домашній клімат-контроль можна здійснити будь-якими способами, використовуючи датчики, термостати та таймери, щоб уникнути клопоту з вашого опалення та охолодження.

Цими пристроями можна керувати дистанційно або самотійно, оскільки вони дізнаються про індивідуальні переваги, які має кожен член будинку, та регулюють нагрівання та охолодження за потребою. Повністю інтегровані розумні будинки здатні забезпечити ідеальну температуру, щоб, коли ви приїжджаєте додому з роботи, ваш будинок був затишним і приємним, без жодного пальця. Домашній клімат-контроль – це технологія, яка швидко вдосконалюється, і нові покоління цих пристроїв пропонують набагато більше, ніж раніше. Ці системи можуть перевірити здоров'я вашого будинку на предмет вологості та інших факторів, покращити власне здоров'я та зробити життя комфортнішим для всіх. Автоматизація кліматичного контролю – це спосіб дистанційного або автоматичного управління нагріванням та охолодженням вашого будинку. Ці розумні пристрої для дому надають детальний звіт про

енерговитрати, температуру та погодні умови, а також працюють на те, щоб постійно опалювати чи охолоджувати ваш будинок до комфортного рівня.

Розумні опалювальні системи для дому покладаються на цілий спектр різних датчиків, термостатів, автоматизованого управління та таймерів роботи. Метою автоматизації кліматичного контролю є зробити свою температуру повністю автоматичною, щоб вам не потрібно було турбуватися про налаштування параметрів або його ввімкнення та вимкнення.

Завдяки домашньому клімат-контролю можна використовувати комп'ютер або смартфон як центральний командний центр та змінювати налаштування за своїм бажанням. Незалежно від того, чи хочете ви обігрівати лише одне приміщення, плануйте, щоб ваші розумні системи охолодження будинку включили кондиціонер за півгодини до приїзду з роботи, або нехай система підтримує оптимальну температуру протягом дня, ці пристрої мають все накрито. Незважаючи на те, що вони є надзвичайно зручними та додають комфорт, системи домашнього кліматичного контролю можуть заощадити власників будинків зовсім небагато грошей. Хоча є початкова інвестиція на придбання пристрою або встановлення більш глибокої системи, вони, як правило, можуть окупитися за економію енерговитрат протягом багатьох років. Дослідження показали, що за допомогою популярного Nest Learning Thermostat економив своїх користувачів від 131 до 145 доларів на рік, ефективно сплачуючи вартість 249 доларів лише за два роки. Оскільки це була модель другого покоління, оновлення останнього пристрою налаштовано, щоб заощадити ще більше. Домашній клімат-контроль – це зручна автоматична функція, яка є в більшості розумних будинків. За допомогою цих автоматичних систем опалення та охолодження ви заощаджуєте багато енергії та думок, які можна більш розумно використовувати в іншому місці.

Сучасні розумні будинки з клімат-контролем можуть зробити повсякденні сценарії настільки зручнішими для їх мешканців. Уявіть, що ви прокидаєтесь кожного ранку в запланований час та ходите до ванної, виявивши, що він уже нагрітий до ідеальної температури, щоб ви могли комфортно насолоджуватися душем.

На сьогоднішній день, існує багато систем розумного дому, які включають в себе функції контролю клімату. Існує дуже багате різноманіття компаній, які виготовляють такі вироби. Комплект систему кліматконтролю включає в себе: диск з програмою, мікроконтролер та різноманітні датчики. Більшість таких виробів можна контролювати лише завдяки ПК, та вдома, що унеможливорює роботу в режимі online, знаходячись не вдома. Також самому контролеру клімату приділено недостатньо уваги, або підсистема контролює тільки дуже вузький спектр показників кліматконтролю. Наприклад, існує дуже багато “розумних” кондиціонерів, термостатів, але все одно вони не можуть об’єднуватись у повноцінну, контролюєму систему.

1.5. Підсистема кліматконтролю на базі Arduino

Ідея складається в тому, щоб створити дистанційно керовану систему автоматизованого розумного будинку для контролю та контролю ступеня температури, відсотка вологості, рівня води. Ця система також може керувати освітленням, використовуючи моторчик, задля відкриття або закриття штор. Розроблена система складається з двох частин: апаратної та програмної. Обладнання складається з багатьох блоків, таких як смартфони, персональні комп’ютери, Arduino Uno та багатьох датчиків. Arduino Uno, що використовується в якості мікроконтролера, є основним апаратним блоком, з’єднаним з роутером, відповідальним за надсилання та отримання команд та статусу через Інтернет. Програмне забезпечення є другою частиною цієї системи;

це код Arduino IDE, який використовується для розробки сторінки веб-браузера. Сторінка браузера працює для контролю та моніторингу цієї системи з веб-сторінки та смартфона. Нарешті, ця система використовує Інтернет-мережу для зв'язку та оповіщення через низьку вартість та велике покриття площі та може контролювати кожен пристрій через веб-браузер в будь-якому місці та в будь-який час.

Висновки до розділу

У даному розділі проведено ознайомлення з системою Розумний дім, його компонентами та складовими. Визначена необхідна компонентна база для створення мікроконтролера для управління кліматом. Виявлено існуючі аналоги та їх недоліки. Описана схема розширеного розумного дому. Визначено об'єкт та предмет дослідження. На основі існуючих аналогів, вибран мікроконтролер Arduino.

РОЗДІЛ 2. КОМПОНЕНТИ ARDUINO UNO TA ADRUINO IDE

2.1. Ознайомлення з Arduino Uno

Плата Arduino Uno – це мікроконтролер на базі ATmega328. Він має 14 цифрових штифтів вводу / виводу, в яких 6 можна використовувати як вихід ШІМ, керамічний резонатор 16 МГц, заголовок ICSP, USB-з'єднання, 6 аналогових входів, роз'єм живлення та кнопку скидання. Тут міститься вся необхідна підтримка, необхідна мікроконтролеру. Для того, щоб розпочати роботу, їх просто підключити до комп'ютера за допомогою кабелю USB або за допомогою адаптера змінного струму або акумулятора. Дошка Arduino Uno відрізняється від усіх інших плат, і вони не будуть використовувати в них чіп-драйвер USB-серійний драйвер FTDI. Він представлений програмою Atmega16U2 (Atmega8U2 до версії R2), запрограмованою як перетворювач USB в послідовний.

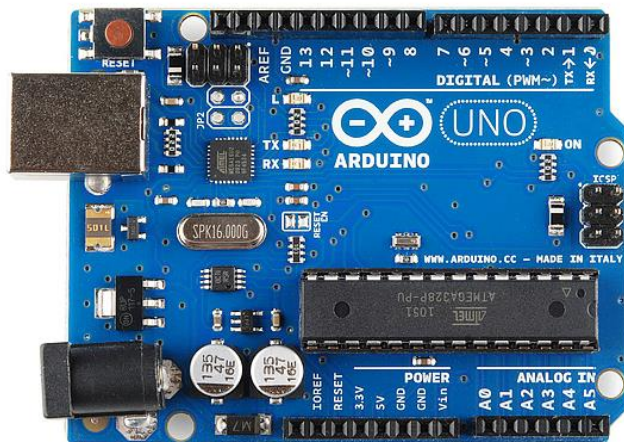


Рисунок 2.1. Вигляд Arduino UNO

2.2. Компоненти Arduino Uno

Обладнання оснащено апаратною платою з відкритим кодом, розробленою навколо 8-бітового мікроконтролера Atmel AVR або 32-бітного Atmel ARM. Поточні моделі складаються з інтерфейсу USB, 6 аналогових входних штифтів

та 14 цифрових вводу-виводу, що дозволяє користувачеві приєднувати різні продовжувачі.

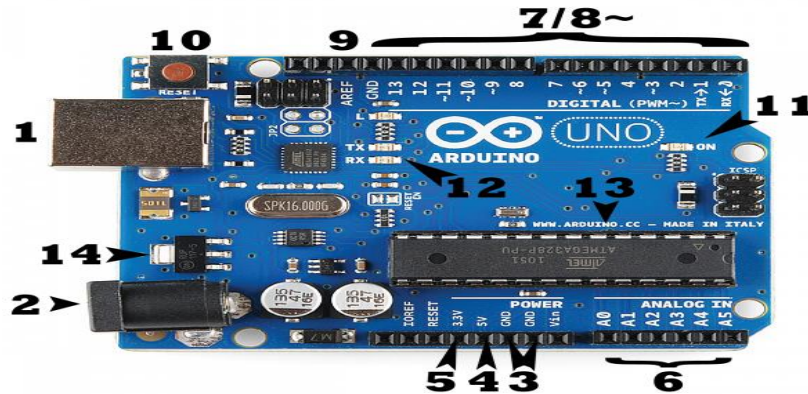


Рисунок 2.2. Елементна база Arduino

Живлення (USB / Barrel Jack)

Кожна плата Arduino потребує способу підключення до джерела живлення. Arduino UNO може житись від USB-кабелю, що надходить від комп'ютера, або від постійного джерела живлення (наприклад, цей), який закінчується в гнізді для ствола. На малюнку над USB-підключенням позначено позначку (1), а розетка бочки - 2 (2). Через USB порт також завантажується код на свою плату Arduino. Не потрібно використовувати джерело живлення більше 20 Вольт, оскільки ви перестанете (і тим самим знищити) свій Arduino. Рекомендована напруга для більшості моделей Arduino становить від 6 до 12 Вольт.

В) Піни: (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Штифти на вашому Arduino – це місця, де ви підключаєте дроти для побудови ланцюга (ймовірно, в поєднанні з дошкою та деяким дротом). Вони, як правило, мають чорні пластикові «заголовки», які дозволяють просто підключити дріт прямо до дошки. має кілька різних типів штифтів, кожен з яких розміщений на дошці та використовується для різних функцій.

5 В (4) та 3,3 В (5): Як ви могли здогадатися, 5 В контактний шнур подає 5 вольт живлення, а 3,3 В – 3,3 вольта живлення. Більшість простих компонентів, які використовуються з Arduino, працюють з 5 або 3,3 вольт;

Аналоговий (6): Площа штифтів під позначкою "Аналоговий вхід" (від А0 до А5 в UNO) – аналогові штифти. Ці штифти можуть зчитувати сигнал з аналогового датчика (як датчик температури) і перетворювати його в цифрове значення, яке ми можемо читати.

Цифровий (7): Попереду від аналогових штифтів знаходяться цифрові штифти (від 0 до 13 в UNO). Ці штифти можна використовувати як для цифрового входу (наприклад, повідомлення про натискання кнопки), так і для цифрового виходу (наприклад, живлення світлодіода).

ШИМ (8): Можливо, ви помітили тильду (~) поруч із деякими цифровими штифтами (3, 5, 6, 9, 10 та 11 в ООН). Ці штифти діють як звичайні цифрові штифти, але їх також можна використовувати для чогось, що називається імпульсно-ширинна модуляція (ШИМ). У нас є навчальний посібник з ШИМ, але наразі подумайте про ці штифти як про можливість імітувати аналоговий вихід (на зразок згасання світлодіода в і поза).

AREF (9): Підставки для аналогових довідок. Більшу частину часу ви можете залишити цю шпильку в спокої. Іноді використовується для встановлення зовнішньої опорної напруги (від 0 до 5 Вольт) як верхньої межі для аналогових вхідних штифтів.

Кнопка перезавантаження

Як і оригінальний Nintendo, Arduino має кнопку скидання (10). Натиснувши на нього, тимчасово підключіть штифт скидання до землі та перезавантажте будь-який код, завантажений на Arduino. Це може бути дуже корисно, якщо ваш код не повторюється, але ви хочете перевірити його кілька разів. На відміну від

оригінального Nintendo, однак, думи по Arduino зазвичай не вирішує жодних проблем.

Індикатор живлення LED

Прямо внизу та праворуч від слова "UNO" на вашій платі є крихітний світлодіод поруч із написом "ON" (11). Цей світлодіод повинен світитися кожного разу, коли ви підключите Arduino до джерела живлення.

TX короткий для передачі, RX – короткий для прийому. Ці позначення з'являються зовсім небагато в електроніці, щоб вказати на штифти, відповідальні за послідовний зв'язок. У нашому випадку на Arduino UNO є два місця, де з'являються TX і RX – один раз цифровими штифтами 0 і 1, а вдруге поруч із індикаторами індикатора TX і RX (12). Ці світлодіоди дадуть нам приємні візуальні індикації, коли наш Arduino отримує або передає дані (наприклад, коли ми завантажуюмо нову програму на плату).

Основна інтегральна схема

Чорна річ з усіма металевими ніжками – це IC, або інтегральна схема (13). Подумайте про це як про мізки нашого Ардуїно. Основний IC на Arduino дещо відрізняється від типу борту до типу борту, але, як правило, від лінії ATmega IC від компанії ATMEL. Це може бути важливо, оскільки вам, можливо, доведеться знати тип IC (разом із типом вашої плати), перш ніж завантажувати нову програму з програмного забезпечення Arduino. Цю інформацію зазвичай можна знайти у письмовій формі у верхній частині ІМС. Якщо ви хочете дізнатися більше про різницю між різними IC, часто читати таблиці даних є хорошою ідеєю.

Регулятор напруги

Регулятор напруги (14) насправді не є тим, з чим можна (або слід) взаємодіяти на Arduino. Але потенційно корисно знати, що він існує і для чого він потрібен. Регулятор напруги робить саме те, що говорить, – він контролює величину напруги, що потрапляє на плату Arduino. Подумайте про це як про якогось воротаря; це відключить зайву напругу, яка може зашкодити ланцюгу. Звичайно, у нього є свої межі, тому не підключайте Arduino до більш ніж 20 вольт.

2.3. Конфігурація пінів

На рис. 2.3. схематично зображено вигляд конфігурації ATmega328.

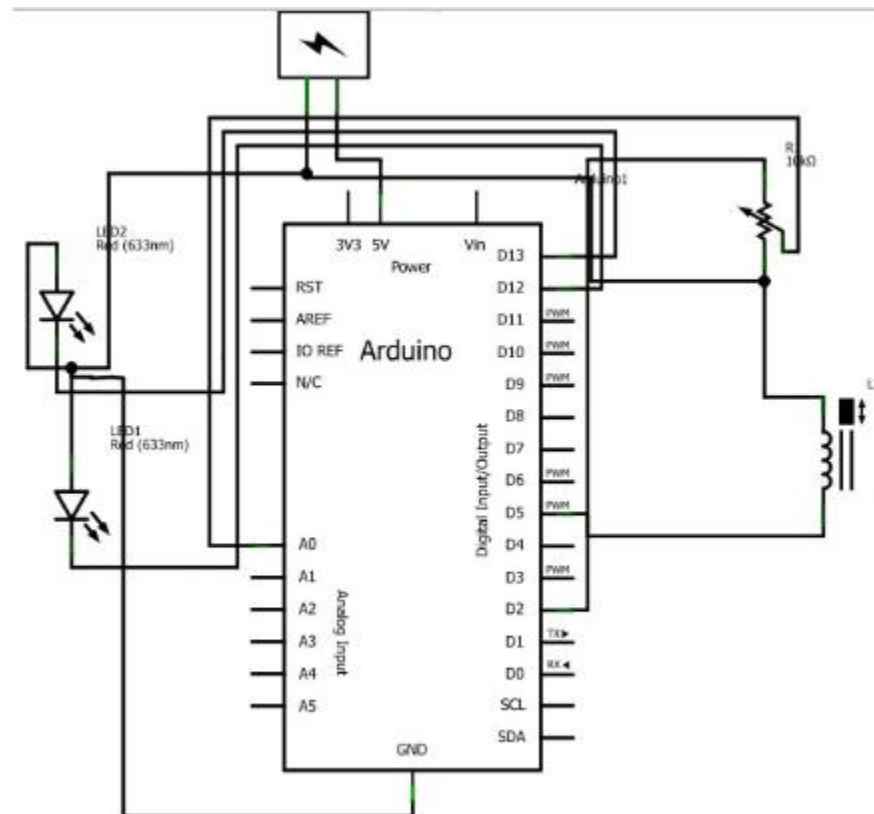


Рисунок 2.3. Вигляд ATmega328

Мікроконтролер
Вольтаж

ATmega328
5V

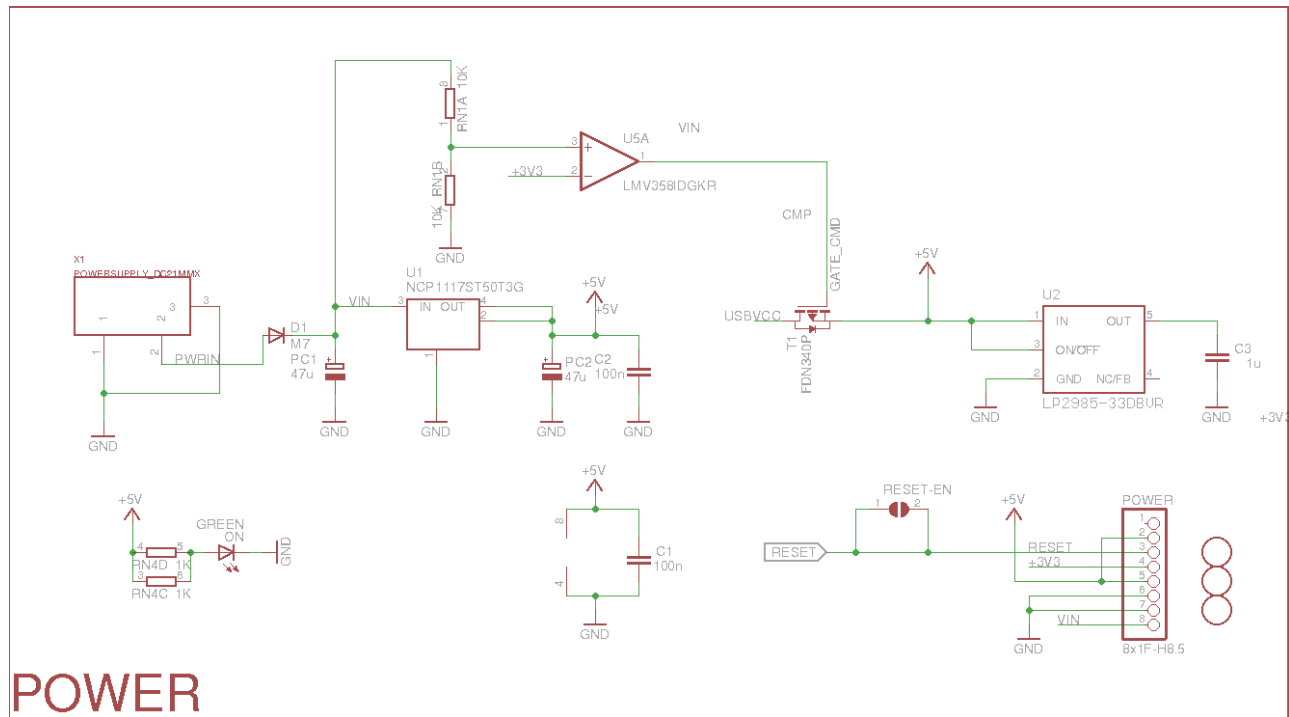


Рисунок 2.5. Схема живлення

Також, на рис.2.6. можна побачити основну схему Arduino Uno.

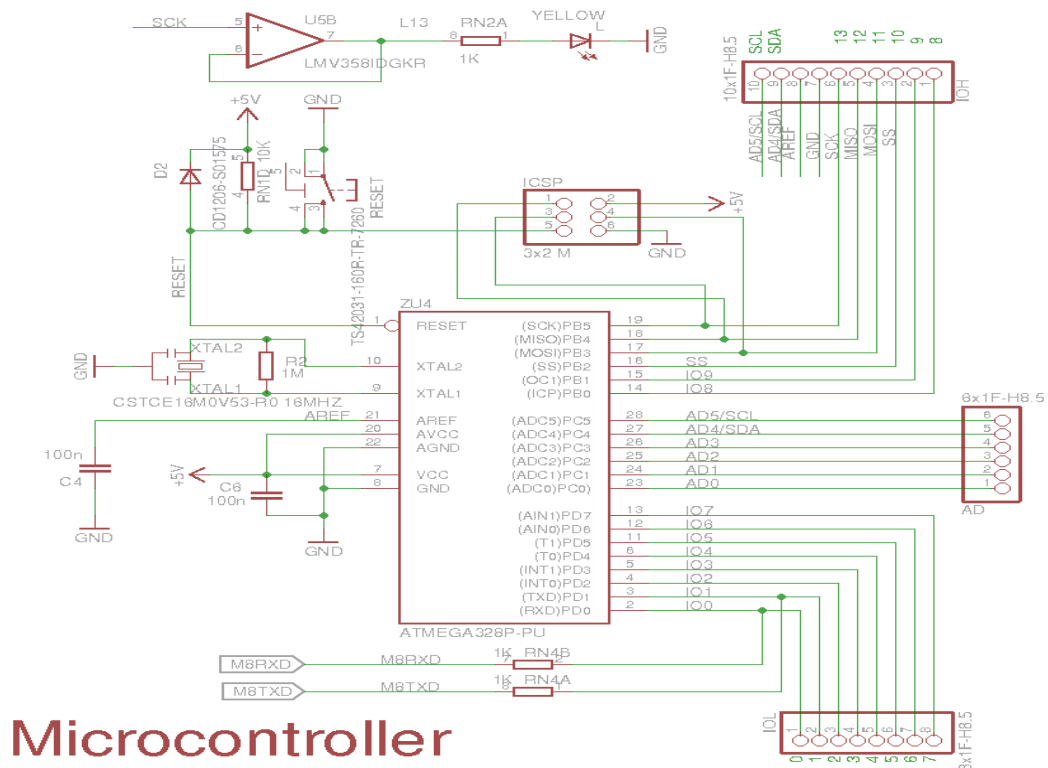


Рисунок 2.6. версія оригінальної схеми Arduino UNO

Входи АЦП

Цей MCU має шість каналів – від PORTC0 до PORTC5 – з 10-бітовим перетворювачем А / D перетворювача. Ці штифти з'єднані з аналоговим заголовком на платі Arduino.

Однією поширеною помилкою є думка про аналоговий вхід як про виділений вхід для функції А / D, оскільки заголовок на платі зазначає "Аналоговий". Реальність полягає в тому, що ви можете використовувати їх як цифрові введення / виведення або А / D.

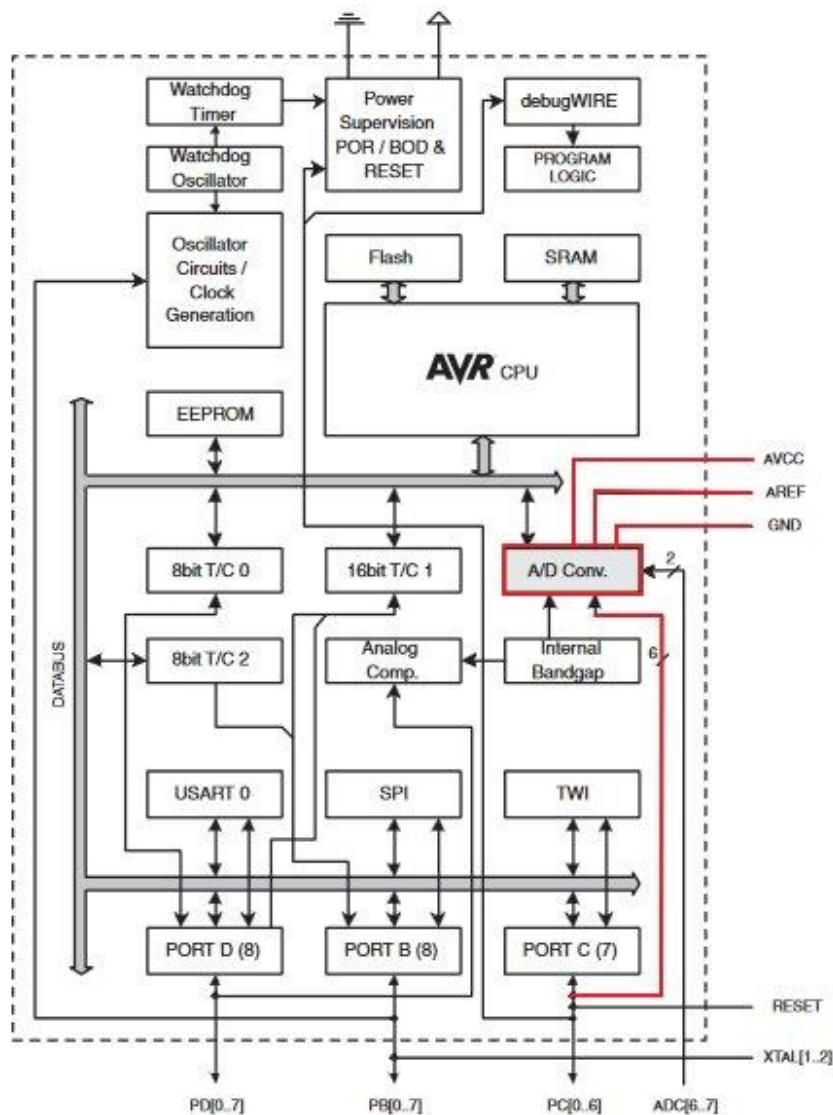


Рисунок 2.9. Блок-схема ATmega168

Як показано на схемі вище (через червоні сліди), штифтами, що відносяться до пристрою A / D, є:

AVCC: штир живлення для A / D блоку.

AREF: вхідний штифт додатково використовується, якщо ви хочете використовувати зовнішню опорну напругу для АЦП, а не внутрішню Vref. Ви можете налаштувати це за допомогою внутрішнього регістра.

Table 24-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Рисунок 2.10. Внутрішні параметри регістра для вибору джерела Vref

2.5. Модулі та інші деталі Arduino

Модуль MOSFET IRF520



Рисунок 2.11. IRF520 MOSFET Driver Module (HCMODU0083)

Модуль датчика температури й вологості DHT11

DHT11 використовує лише один сигнальний провід для передачі даних Arduino. Потужність відбувається від окремих 5В і заземлення. Підтягуючий резистор 10 К Ом потрібен між сигнальною лінією та лінією 5 В, щоб рівень сигналу залишався високим за замовчуванням.



Рисунок 2.13. Датчик DHT11

Існує дві різні версії DHT11, на які ви можете зіткнутися. Один тип має чотири штифти, а другий тип має три штифти і кріпиться до невеликої плати. Версія, встановлена на друкованій платі, приємна тим, що вона включає поверхневий підтягуючий резистор 10 К Ом для сигнальної лінії. Ось пін-виходи для обох версій:

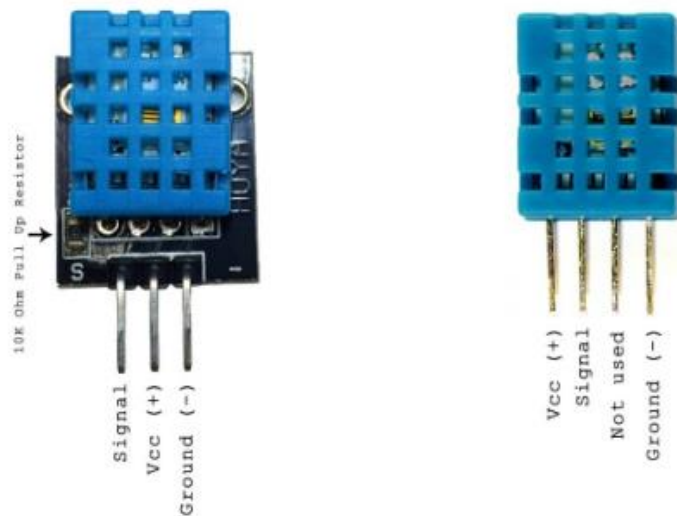


Рисунок 2.14. Пін-виходи

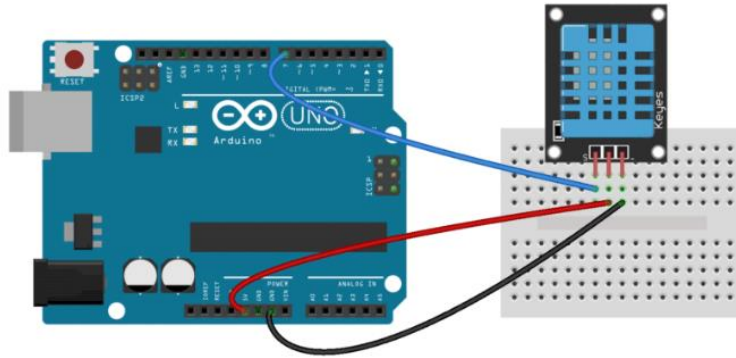


Рисунок 2.15. Схема підключення DHT11

Одно-канальний модуль реле

Він має штифтовий заголовок 1×3 (2,54 мм) для підключення живлення (5 В та 0 В) та для управління реле. Шпильки позначені на друкованій платі:



Рисунок 2.16. Реле

GND - підключіть 0V до цього контакту.

SIG - Керує цим реле, активно Низький! Реле вмикається, коли цей вхід опуститься нижче приблизно 2,0 В

VCC - підключіть 5V до цього штифта. Використовується для живлення оптронів

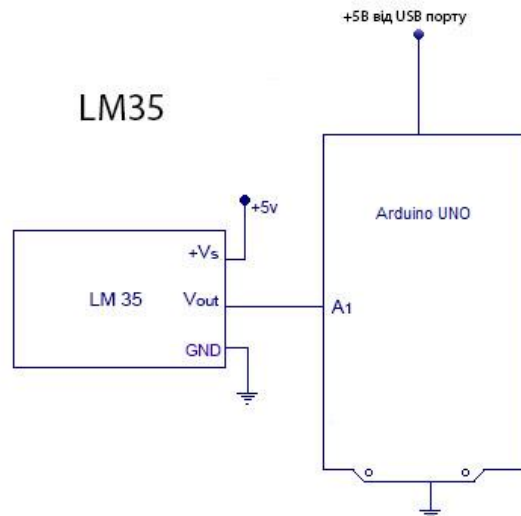


Рисунок 2.18. Схема підключення lm35

Підключіть LM35 до Arduino uno, як показано на схемі. + 5 В для LM35 можна взяти з + 5 В на штирі arduino uno. Також заземлений штифт LM35 можна підключити до штифта GND arduino uno. Підключіть Vout (аналоговий вихід з LM35) до будь-якого аналогового входного штифта arduino uno.

Фоторезистор GL5516

Фоторезистор – це резистор, виготовлений з напівпровідникового матеріалу, і провідність змінюється з варіацією яскравості.



Рисунок 2.19. Датчик GL5516

Фоторезистор може бути виготовлений з різними фігурами і підсвічений область, заснована на цій характеристиці. Фоторезистор широко використовується в багатьох галузях промисловості, таких як іграшки, світильники, фотоапарат тощо.

WiFi модуль ESP8266

Підключіть вихід Arduino 3v3 (3.3V) до червоної лінії на дошці. ESP8266 працює з 3,3 В, а не 5 В, тому це необхідно.



Рисунок 2.20. WiFi модуль - esp8266-07

Якщо ви хочете підключити інші компоненти, які використовують 5В, ви можете підключити вихід 5В до іншої червоної лінії дошки, просто переконайтеся, що ви не з'єднали ці два.

1. Підключіть GND (землю) до синьої лінії;
2. Підключіть штифт RES або RESET до синьої лінії. Коли ви заземлите контактний контакт, Arduino працює як тупий USB-послідовний роз'єм, про що ми хочемо поговорити з ESP8266.
2. Підключіть штифт RXD Arduino до штифта RX ESP8266 (жовтий колір на малюнку);
3. Підключіть штифт TXD Arduino до штифта TX ESP (зелений колір на малюнку). Зазвичай, коли ми хочемо, щоб дві речі розмовляли між собою за допомогою послідовного зв'язку, ми підключаємо штифт TX одного до RX іншого (відправка йде на отримання та навпаки). Тут ми не маємо розмови про Arduino з ESP8266, хоча наш комп'ютер розмовляє з ним через Arduino;

4. Підключіть штифт GND ESP до синьої лінії, а штифт VCC – до червоної;
5. Нарешті CH_PD переходить до червоної лінії, нібито це не спрацює, якщо ви не підключите це.

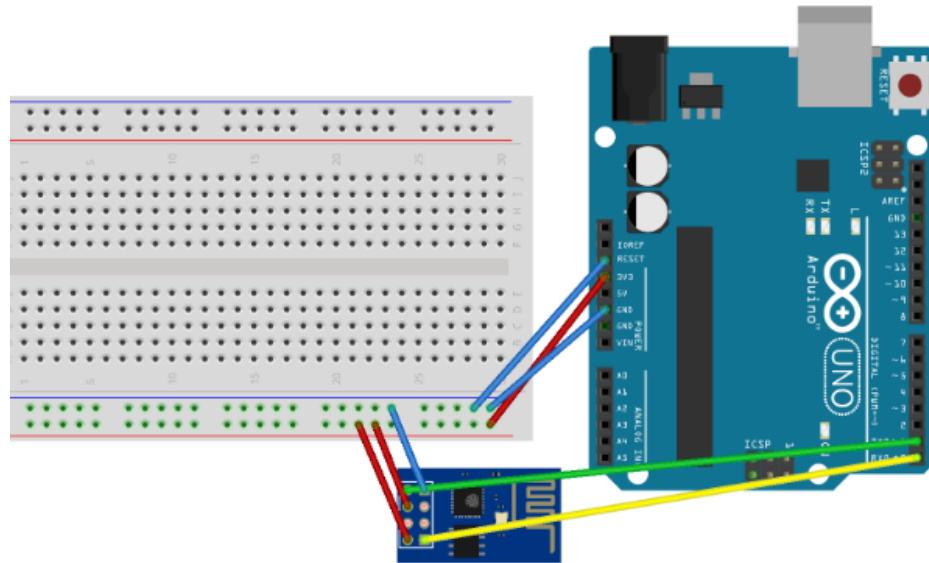


Рисунок 2.21. Схема підключення модуля esp8266-07

Електромагнітний клапан 1/2 "12В нормально-закритий соленоїд

Використання Arduino для управління електромагнітним клапаном є просто випадком встановлення високого штифта за відповідний проміжок часу.



Рисунок 2.22. Соленоїдний клапан

Однак є застереження: соленоїд працює при різній напрузі до Ардуїно, і ви не можете їх безпосередньо з'єднати. У цьому випадку транзистор TIP120 використовується як міст.

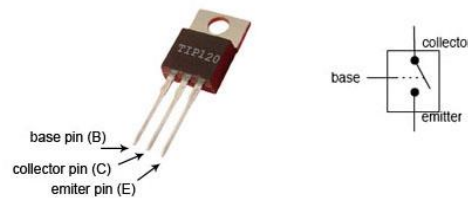


Рисунок 2.23. Транзистор TIP120

Діод (в даному випадку 1N4007), підключений до соленоїду, дозволяє струму текти лише в одному напрямку.

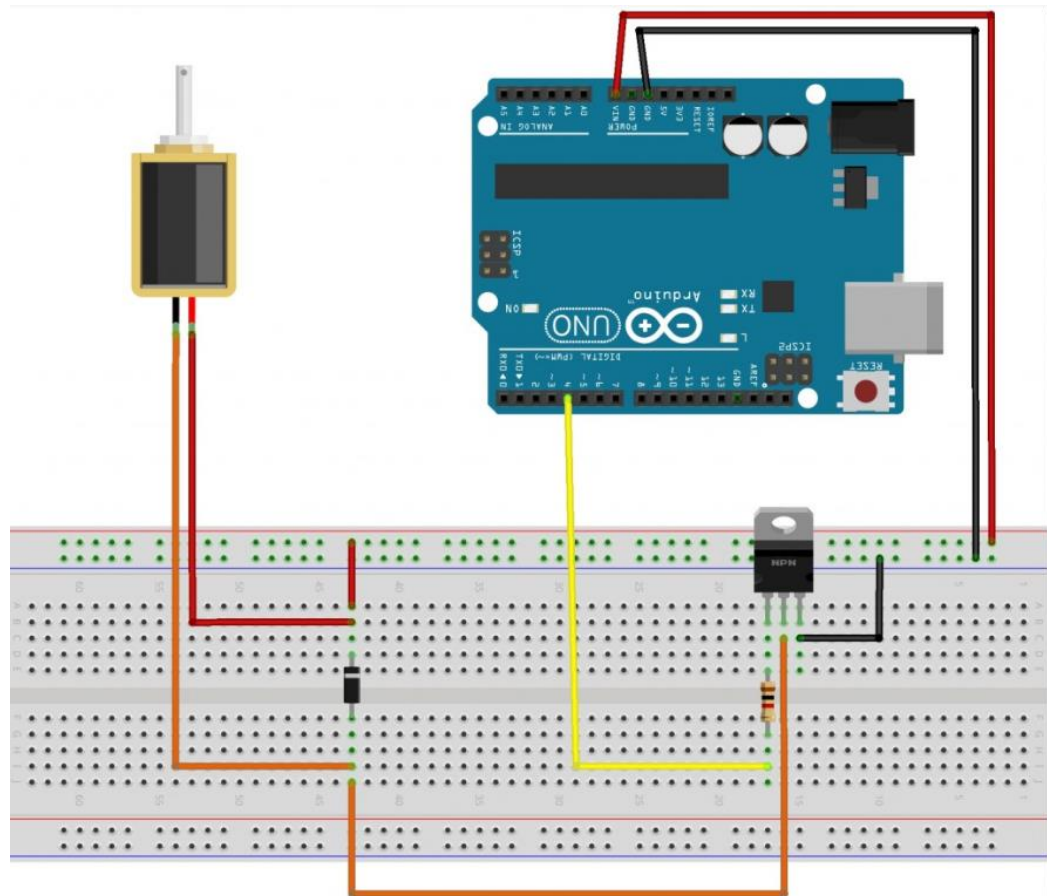


Рисунок 2.24. Підключення соленоїда до Arduino

При відключенні струму соленоїд намагається продовжити струм. Діод подає цей струм назад в соленоїд, поки він не розсіюється.

Мотор постійного струму 9V

Двигуни постійного струму насправді дуже схожі на сервомотори безперервного обертання. Щодо останнього, токова моторна технологія набагато простіша, і, як наслідок, ці мотори дешевші, ніж сервоприводи.

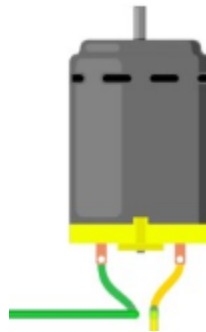


Рисунок 2.25. Мотор 9В

Як пропонує їх ім'я, двигунам постійного струму для запуску потрібне лише джерело постійного струму, тому не потрібно використовувати жоден штировий ШІМ. У них просто два дроти, які потрібно підключити до джерела напруги. Напрямок обертання залежить від способу підключення проводів до джерела напруги. Повернення полярності потужності змушує мотор працювати в зворотному напрямку. Швидкість двигуна постійного струму залежить від прикладеної напруги: чим вище напруга, тим більша швидкість. Технічні характеристики двигуна говорять про те, в якому діапазоні напруги він працює: нижче мінімальної напруги мотор, ймовірно, взагалі не буде рухатися або вони рухаються з великими труднощами; вище максимальної напруги, ви серйозно ризикуєте пошкодити її.

Однак у деяких випадках ця дуже проста схема не може працювати: явно вона не працює, якщо 5В недостатньо для роботи двигуна; до того ж,

максимальний струм, який може подати штифт Arduino, становить 40 мА (але рекомендується не перевищувати 20 мА), і його може бути недостатньо для роботи двигуна. У цих випадках потрібно зовнішнє джерело живлення. Ви не можете використовувати Arduino як джерело живлення, але ви можете використовувати його як драйвер двигуна, керуючи зовнішнім джерелом живлення, яке використовується для живлення двигуна. Припустимо, наприклад, що для роботи мотора вам потрібно щонайменше 9В. Ви можете використовувати Arduino для здійснення свого роду перемикача, який залишається відкритим до тих пір, поки для цифрового штифта не буде встановлено значення HIGH. Як тільки цей штифт перевертається з НИЗКОГО в ВИСОК, перемикач закривається, а двигун підключається до джерела живлення.

Звичайно, для цього можна використовувати реле, але ми проілюструємо використання транзистора для отримання того ж результату. Розуміння того, як використовувати транзистори, – це завжди хороша ідея, навіть якщо ви їх не збираєтеся використовувати: знання – це сила!

Транзистори можна розглядати як своєрідні перемикачі. Насправді транзистори можуть виступати як підсилювачами, так і комутаторами, залежно від того, як вони поляризовані. Якщо ви хочете зрозуміти, як транзистори працюють не технічним чином, ви можете прочитати мою статтю, опубліковану на темі "Вчитель фізики" (Джованні Органтіні, "Модель туалету для транзистора", Phys. Teach. 50, 221 (2012); <http://dx.doi.org/10.1119/1.3694073>). Щоб зрозуміти цю публікацію, вам не потрібно розуміти, як працює транзистор: достатньо знати, що вони поведуться як комутатори. Перемикач з'єднує транзисторний випромінювач з його колектором, і його робота керується його базою. Розглянемо наступну схему.

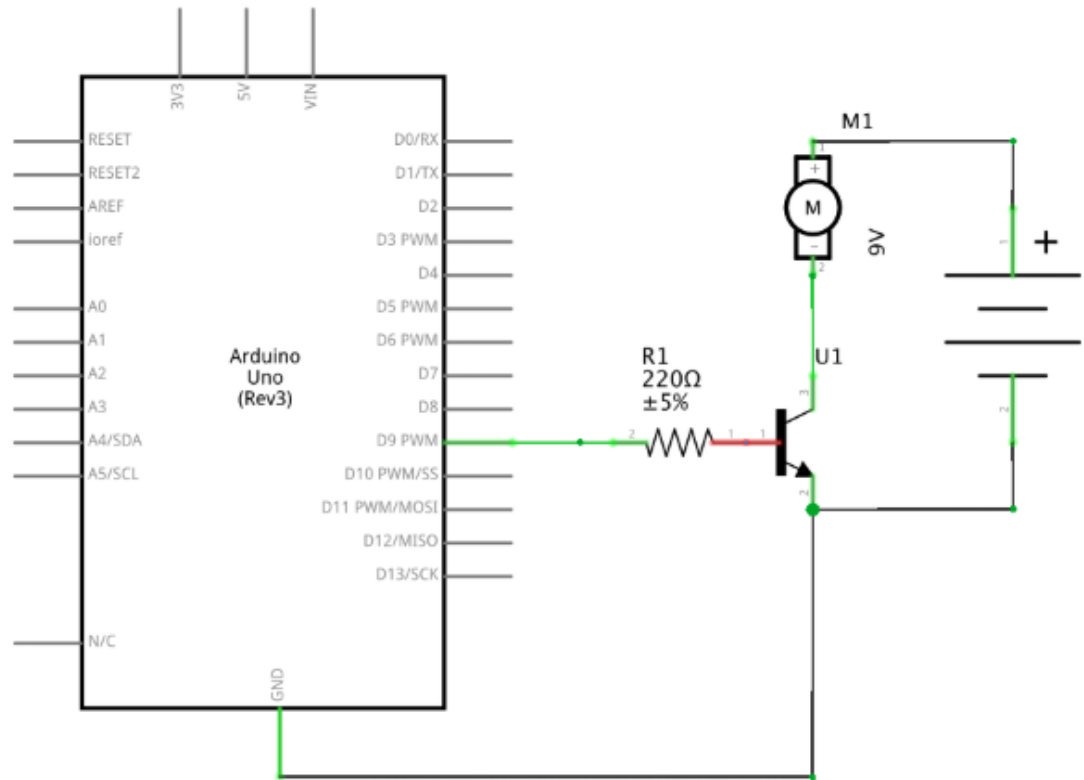


Рисунок 2.26. Схема мотора

Плата Arduino використовується просто для подачі цифрового сигналу на заданий контакт (контакт 9 у прикладі). Цей штифт з'єднаний з резистором (значення якого не дуже важливо: його мета полягає лише в обмеженні струму, що витікає з штифта), який, в свою чергу, підключається до основи транзистора. Якщо до основи транзистора не надходить струм, то, як кажуть, транзистор перебуває в забороні і виступає як відкритий перемикач між випромінювачем і колектором. У нашій схемі двигун підключений до 9В акумулятора через транзистор, але ви можете уявити, що транзистор замінений відкритим перемикачем. Земля Arduino є спільною з мінус 9В.

Як тільки ви зробите достатню кількість струму, що протікає в базу транзистора, транзистор переходить в насиченість і діє як закритий вимикач. У цьому випадку транзисторний випромінювач (нижній відвід зі стрілкою на схемі) виявляється в короткому замиканні зі своїм колектором (верхній провід,

підключений до двигуна). У цьому випадку струм може надходити від акумулятора і живити двигун, який починає працювати.

Нижче ви можете побачити модель Фріцзінга пристрою, де ви бачите, що основою транзистора є його центральний провід; випромінювач знаходиться зліва, а колектор – справа.

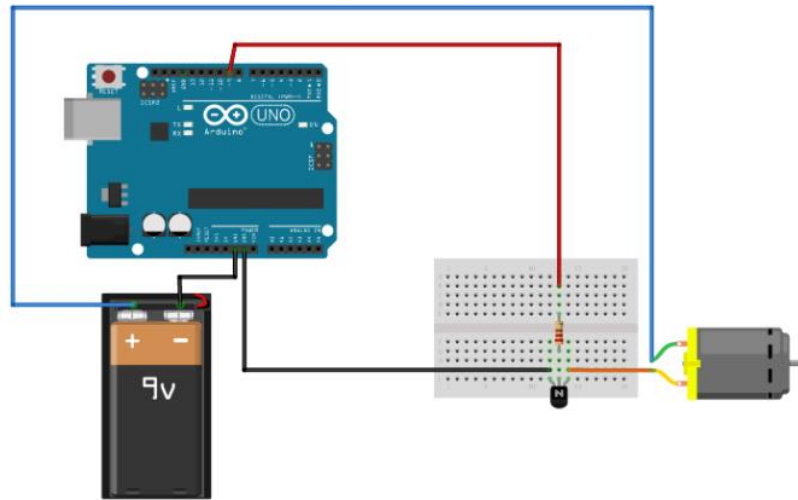


Рисунок 2.27. Схема підключення мотора до Arduino

Резистори

Середовище та технології

Для даного пристрою використане середовище:

Використані такі технології як:

1. HTML
2. CSS
3. Java Sript
- 2.6. Arduino IDE

Програми Arduino написані в середовищі інтегрованого розвитку Arduino (IDE). Arduino IDE – це спеціальне програмне забезпечення, що працює у вашій системі, яке дозволяє писати ескізи (синонім програми на мові Arduino) для різних плат Arduino. Мова програмування Arduino заснований на дуже простій

апаратній мові програмування, що називається обробкою, яка схожа на мову C. Після того, як ескіз записаний в IDE Arduino, його слід завантажити на дошку Arduino для виконання.

Першим кроком у програмуванні плати Arduino є завантаження та встановлення ID Arduino IDE. Arduino IDE з відкритим кодом працює на Windows, Mac OS X та Linux. Завантажте програмне забезпечення Arduino (залежно від вашої ОС) з офіційного веб-сайту та дотримуйтесь інструкцій із встановлення.

Підготовка та виконання

Кожен блок має набір операторів, укладених у фігурні дужки:

```
void setup( )
{
  statements-1;
  .
  .
  .
  statement-n;
}
void loop ( )
{
  statement-1;
  .
  .
  .
  statement-n;
}
```


Тут *setup ()* – це блок підготовки, а цикл *()* – блок виконання.

Функція настройки першою виконується при виконанні програми, і ця функція викликається лише один раз. Функція налаштування використовується для ініціалізації штифтових режимів та початку послідовного зв'язку. Ця функція повинна бути включена, навіть якщо немає жодних операторів, які потрібно виконати.

```
void setup ( )
{
pinMode (pin-number, OUTPUT); // set the 'pin-number' as output
pinMode (pin-number, INPUT); // set the 'pin-number' as output
}
```

Після того, як функція 3 буде виконана, блок виконання працює далі. Блок виконання розміщує оператори, такі як читання входів, запуск виходів, перевірка умов тощо.

У наведеному вище прикладі функція *loop ()* є частиною блоку виконання. Як випливає з назви, функція *loop ()* виконує набір операторів (укладених у фігурні дужки) кілька разів.

```
Void loop ( )
{
digitalWrite (pin-number,HIGH); // turns ON the component connected to
'pin-number'
delay (1000); // wait for 1 sec
digitalWrite (pin-number, LOW); // turns OFF the component connected to
'pin-number'
delay (1000); //wait for 1sec
}
```

Етапи в побудові з'єднання з дошкою:

Крок 1. Підключіть Arduino до системи Windows / Mac / Linux за допомогою кабелю USB;

Крок 2: Підключіть 13-й цифровий штифт Arduino до позитивної позиції потужності макетної плати, а GND – до негативної;

Крок 3: Підключіть клему позитивної потужності до клемної смуги через резистор 1 К Ом;

Крок 4: Зафіксуйте світлодіод на порти нижче резисторного з'єднання в клемній смугі;

Крок 5: Замкніть ланцюг, підключивши катод (короткий акорд) світлодіода до смуги негативної потужності на дошці.

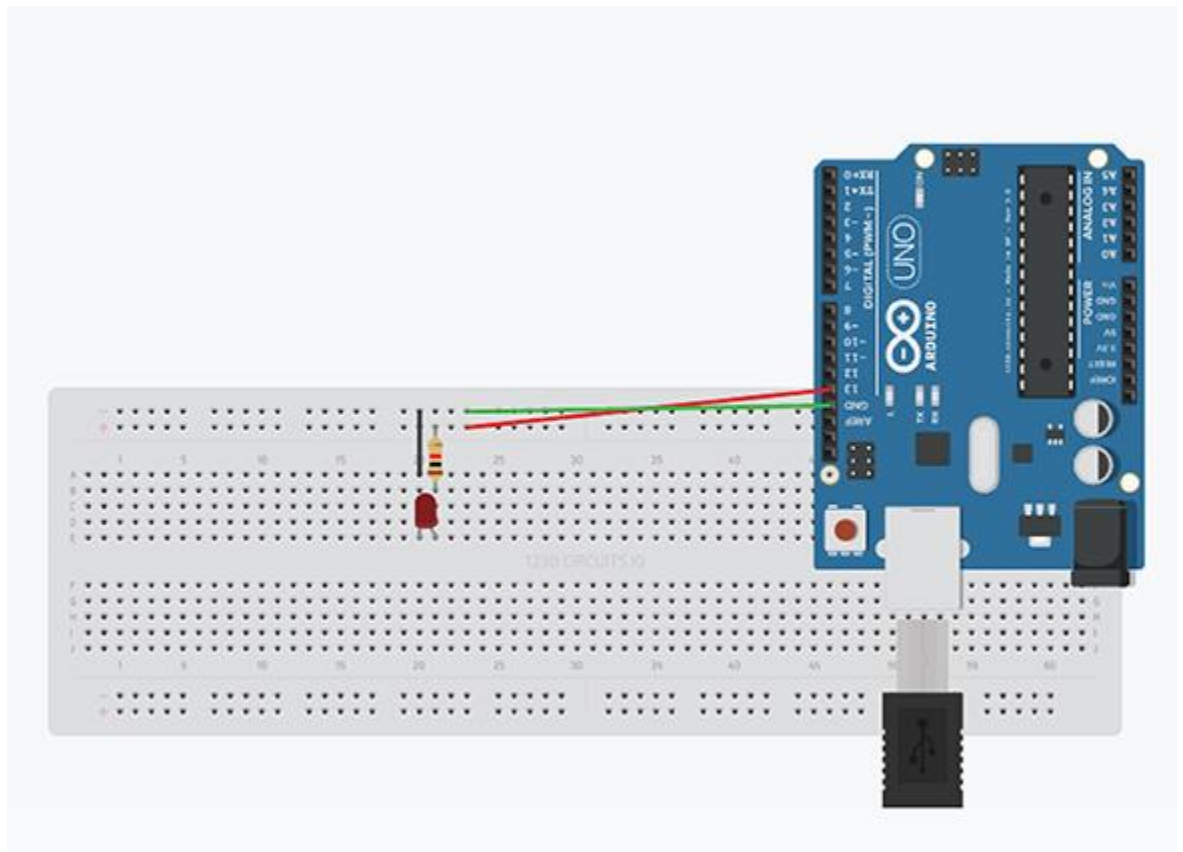


Рисунок 2.28. Приклад підключення світлодіода до Arduino

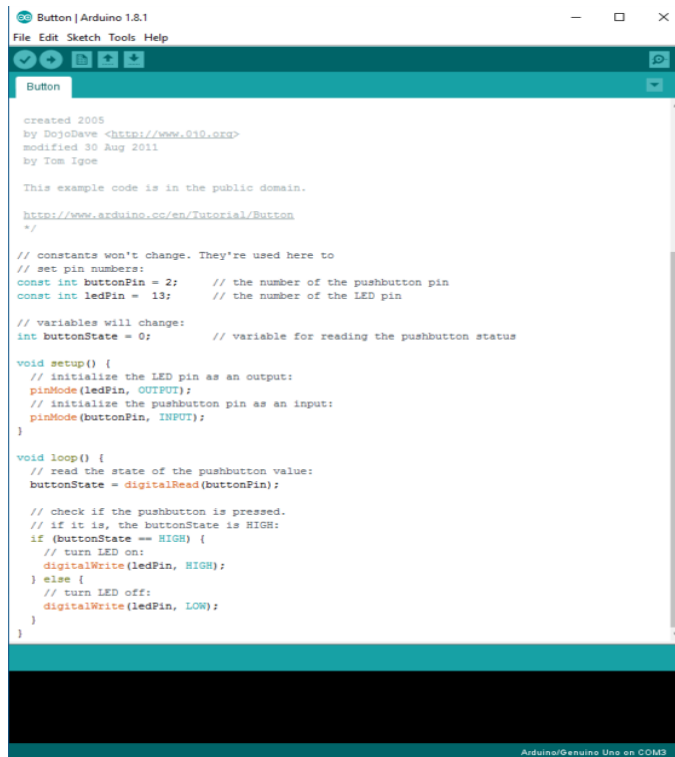


Рисунок 2.29. Arduino IDE

Тепер ми можемо повністю та безперешкодно працювати з мікроконтролером.

Бібліотека ESP8266WiFi

Бібліотека Wi-Fi для ESP8266 була розроблена на базі SDK ESP8266, використовуючи конвенції іменування та загальну філософію функціонування бібліотеки Wi-Fi Wi-Fi Arduino. З часом багатство функцій Wi-Fi перенесено з ESP9266 SDK на esp8266 / Arduino з переростання бібліотеки Wi-Fi у Arduino, і стало очевидним, що нам потрібно надати окрему документацію щодо того, що нового та додаткового.

Ця документація допоможе вам ознайомитися з кількома класами, методами та властивостями бібліотеки ESP8266WiFi. Якщо ви новачок у C++ та Arduino, не хвилюйтеся. Ми почнемо з загальних понять, а потім перейдемо до детального опису членів кожного конкретного класу, включаючи приклади використання.

Область функціонування, яку пропонує бібліотека ESP8266WiFi, досить обширна, і тому цей опис розбито на окремі документи, позначені: `arrow_right` .

Сподіваємось, ви вже знаєте, як завантажити ескіз Blink.ino на модуль ESP8266 і отримати світловий індикатор. Якщо ні, будь ласка, використовуйте цей підручник від Adafruit або інший чудовий підручник, розроблений Sparkfun.

Щоб підключити модуль ESP до Wi-Fi (наприклад, підключення мобільного телефону до гарячої точки), вам потрібно лише пару рядків коду:

```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  Serial.println();

  WiFi.begin("network-name", "pass-to-network");

  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.print("Connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {}
```

Рисунок 2.30. Приклад підключення модулю ESP

У рядку `WiFi.begin("network-name", "pass-to-network")` замініть *network-name* та *pass-to-network* на ім'я та пароль мережі Wi-Fi, до якої ви хочете підключитися. Потім завантажте цей ескіз до модуля ESP і відкрийте послідовний монітор. Ви повинні побачити щось на кшталт:

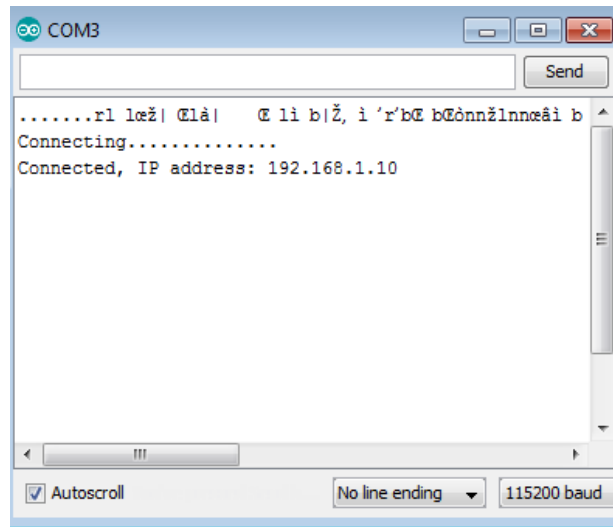


Рисунок 2.31. Скріншот успішного підключення

У першому рядку ескізу, `#include <ESP8266WiFi.h>`, ми включаємо бібліотеку ESP8266WiFi. Ця бібліотека надає специфічні підпрограми Wi-Fi для ESP8266, які ми закликаємо підключитися до мережі.

Фактичне підключення до Wi-Fi ініціюється за допомогою виклику:

```
WiFi.begin("network-name", "pass-to-network");
```

Рисунок 2.32. Функція підключення до WIFI

Процес підключення може зайняти пару секунд, і ми перевіряємо, чи завершилось це в наступному циклі:

```
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
```

Рисунок 2.33. Цикл перевірки підключення

Цикл `while ()` триматиме циклічність, поки `WiFi.status ()` не відрізняється від `WL_CONNECTED`. Цикл вийде, лише якщо стан зміниться на `WL_CONNECTED`.

В останньому рядку буде роздруковано IP-адресу, призначену DHCP модулю ESP:

```
Serial.println(WiFi.localIP());
```

Рисунок 2.34. Функція виведення IP адреси на екран

Якщо ви не бачите останнього рядка, але просто все більше та більше крапок, ймовірно, ім'я або пароль до мережі Wi-Fi введено неправильно в ескіз. Перевірте ім'я та пароль, підключившись з нуля до цієї Wi-Fi мережі за допомогою ПК чи мобільного телефону.

Це все, що потрібно для підключення ESP8266 до Wi-Fi. У наступних розділах ми пояснимо, які класні речі може зробити ESP, коли він підключений.

Пристрої, що підключаються до мереж Wi-Fi, називаються станціями (STA). Підключення до Wi-Fi забезпечується точкою доступу (AP), яка виступає центром для однієї або декількох станцій. Точка доступу на іншому кінці підключена до дротової мережі. Точка доступу зазвичай інтегрується з маршрутизатором для забезпечення доступу з мережі Wi-Fi до Інтернету. Кожна точка доступу розпізнається SSID (ідентифікатор набору послуг), що по суті - це назва мережі, яку ви вибираєте під час підключення пристрою (станції) до Wi-Fi.

Модулі ESP8266 можуть працювати як станція, тому ми можемо підключити її до мережі Wi-Fi. Він також може працювати як м'яка точка доступу (soft-AP), щоб створити власну мережу Wi-Fi. Коли модуль ESP8266 працює як м'яка точка доступу, ми можемо підключити інші станції до модуля ESP. ESP8266 також може працювати як станція, так і режим м'якої точки доступу. Це забезпечує можливість будівництва, напр. сітчасті мережі.



Рисунок 2.35. Схема мережі підключення мікроконтролера до Інтернету

Бібліотека ESP8266WiFi забезпечує широку колекцію методів (функцій) C++ та властивостей для налаштування та роботи модуля ESP8266 у режимі станції та / або програмного забезпечення точки доступу.

Бібліотека ESP8266WiFi розбита на кілька класів. У більшості випадків при написанні коду користувач не переймається цією класифікацією. Ми використовуємо його для розбиття опису цієї бібліотеки на більш керовані частини.



Рисунок 2.36. Складові частини бібліотеки ESP8266WiFi

Режим станції (STA) використовується для підключення модуля ESP до мережі Wi-Fi, встановленої точкою доступу.



Рисунок 2.37. ESP8266WiFi в режимі “роздачі”

Клас станції має кілька функцій для полегшення управління Wi-Fi-з'єднанням. У разі втрати з'єднання, ESP8266 автоматично підключиться до останньої використаної точки доступу, як тільки вона буде знову доступна. Те саме відбувається при перезавантаженні модуля. Це можливо, оскільки ESP зберігає облікові дані до останньої використаної точки доступу у флеш (енергонезалежній) пам'яті. Використання збережених даних ESP також відновиться, якщо ескіз був змінений, але код не змінює режим Wi-Fi або облікові дані.

Точка доступу (AP) – це пристрій, який забезпечує доступ до мережі Wi-Fi до інших пристроїв (станцій) та з'єднує їх з дротовою мережею. ESP8266 може забезпечити подібну функціональність, за винятком того, що він не має інтерфейсу до дротової мережі. Такий режим роботи називається м'якою точкою доступу (soft-AP). Максимальна кількість станцій, які можуть бути одночасно підключені до програмного забезпечення AP, може бути встановлено від 0 до 8, але за замовчуванням – 4.



Рисунок 2.38. ESP8266WiFi в режимі вільного доступу

У режимі м'якої AP часто використовується і проміжний крок до підключення ESP до Wi-Fi у станційному режимі. Це коли SSID та пароль до такої мережі не відомі заздалегідь. ESP спочатку завантажується в режимі soft-AP, тому ми можемо підключитися до нього за допомогою ноутбука чи мобільного телефону. Тоді ми можемо надати облікові дані цільовій мережі. Потім ESP перемикається на станційний режим і може підключитися до цільового Wi-Fi.

Ще одне зручне застосування в режимі soft-AP - це налаштування сітчастих мереж. ESP може працювати як в режимі Soft-AP, так і в станції, тому він може діяти як вузол сітчастої мережі.

Щоб підключити мобільний телефон до гарячої точки, зазвичай ви відкриваєте додаток налаштувань Wi-Fi, перераховуєте доступні мережі та вибираєте потрібну гарячу точку. Потім введіть пароль (чи ні) і ви перебуваєте. Ви можете зробити те ж саме з ESP. Функціональність сканування та переліку доступних мереж у діапазоні реалізується Класом сканування.

Клас Клієнт створює клієнтів, які можуть отримувати доступ до послуг, що надаються серверами, з метою надсилання, отримання та обробки даних.



Рисунок 2.39. ESP8266 в режимі Client

AxTLS Client Secure - це розширення класу клієнтів, де з'єднання та обмін даними з серверами здійснюється за допомогою захищеного протоколу. Він підтримує TLS 1.1. TLS 1.2 не підтримується. Захищені програми мають додаткову пам'ять (і обробку) накладні витрати через необхідність запускати алгоритми криптографії. Чим сильніше ключ сертифіката, тим більше накладних витрат. На практиці неможливо запустити більше одного безпечного клієнта одночасно. Проблема стосується оперативної пам'яті, яку ми не можемо додати; зазвичай не проблема розміру флеш-пам'яті.

BearSSL :: WiFiClientSecure та BearSSL :: WiFiServerSecure – це розширення стандартних класів клієнтів та серверів, де з'єднання та обмін даними з серверами та клієнтами за допомогою захищеного протоколу. Він підтримує TLS 1.2, використовуючи широкий спектр сучасних шифрів, хешей та ключових типів.



Рисунок 2.40. ESP8266 в режимі Client Secure

Безпечним клієнтам та серверам потрібні значні обсяги додаткової пам'яті та обробки для включення їх криптографічних алгоритмів. Як правило, лише один захищений клієнт або серверне з'єднання одночасно може бути оброблений, враховуючи невелику оперативну пам'ять, наявну в ESP8266, але існують методи зниження цієї вимоги ОЗУ, детально описані у відповідних розділах.

Клас сервера створює сервери, які надають функціональні можливості іншим програмам або пристроям, званім клієнтами.

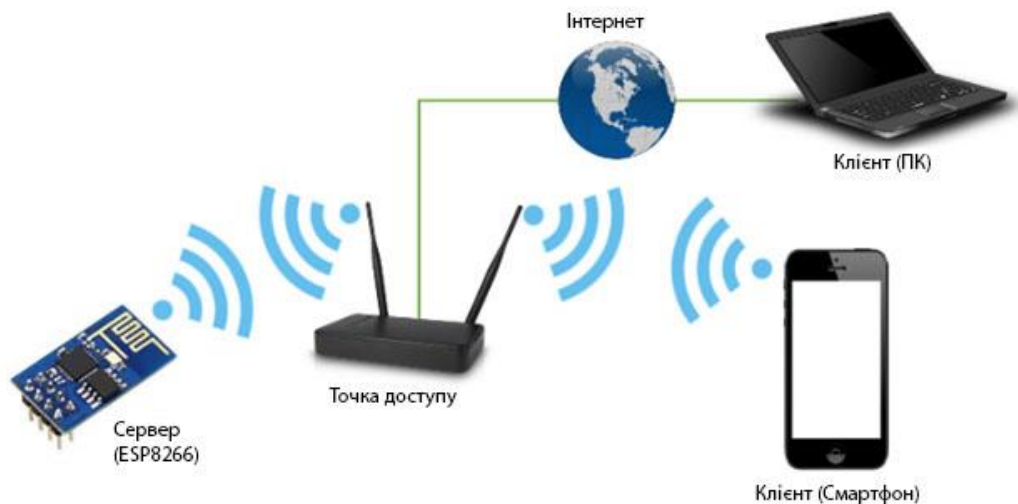


Рисунок 2.41. ESP8266 в режимі серверу

Клієнти підключаються до сервера для надсилання та отримання даних та доступу до наданої функціональності.

Клас UDP дозволяє надсилати та отримувати повідомлення протоколу User Datagram Protocol (UDP). UDP використовує просту модель передачі "пожежа та забудь" без гарантії доставки, замовлення або дублювання захисту. UDP надає контрольні суми на цілісність даних та номери портів для адреси різних функцій у джерелі та в адресі дейтаграми.

Існує кілька функцій, запропонованих SDK ESP8266, і вони відсутні у бібліотеці Wi-Fi Arduino. Якщо така функція не вписується в один із розглянутих вище класів, швидше за все, це буде в Generic Class. Серед них – обробник для керування подіями Wi-Fi, такими як підключення, відключення або отримання IP-адреси, зміни режиму Wi-Fi, функції управління режимом сну модуля, ім'я хоста до роздільної здатності IP-адреси тощо.

Увімкнення діагностики Wi-Fi

За замовчуванням діагностичний вихід з бібліотек Wi-Fi вимикається під час виклику `Serial.begin`. Щоб знову увімкнути вихід налагодження, зателефонуйте `Serial.setDebugOutput (true)`. Щоб замість цього перенаправити вихід налагодження на `Serial1`, зателефонуйте на `Serial1.setDebugOutput (true)`. Детальнішу інформацію щодо діагностики за допомогою послідовних портів див. У документації.

Нижче наведено приклад виводу для зразкового ескізу, обговорюваного у швидкому старті вище із `Serial.setDebugOutput (true)`:

```
Connecting scandone
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
aid 1
cnt

connected with sensor-net, channel 6
dhcp client start...
chg_B1:-40
...ip:192.168.1.10,mask:255.255.255.0,gw:192.168.1.9
.
Connected, IP address: 192.168.1.10
```

Рисунок 2.42. Приклад виводу

Цей самий ескіз без `Serial.setDebugOutput (true)` буде друкувати лише наступне:

```
Connecting....
Connected, IP address: 192.168.1.10
```

Рисунок 2.43. Вивід на екран

Якщо ви хочете детально проаналізувати, що знаходиться всередині бібліотеки ESP8266WiFi, перейдіть безпосередньо до папки ESP8266WiFi в сховищі esp8266 / Arduino на GitHub.

Щоб зробити аналіз простішим, а не заглядати в окремі заголовки чи вихідні файли, використовуйте один із безкоштовних інструментів для автоматичного створення документації. Індекс класів у розділі Опис класу вище був підготовлений за короткий час із застосуванням великого Doxygen, що є фактичним стандартним інструментом для створення документації з анотованих джерел C++.

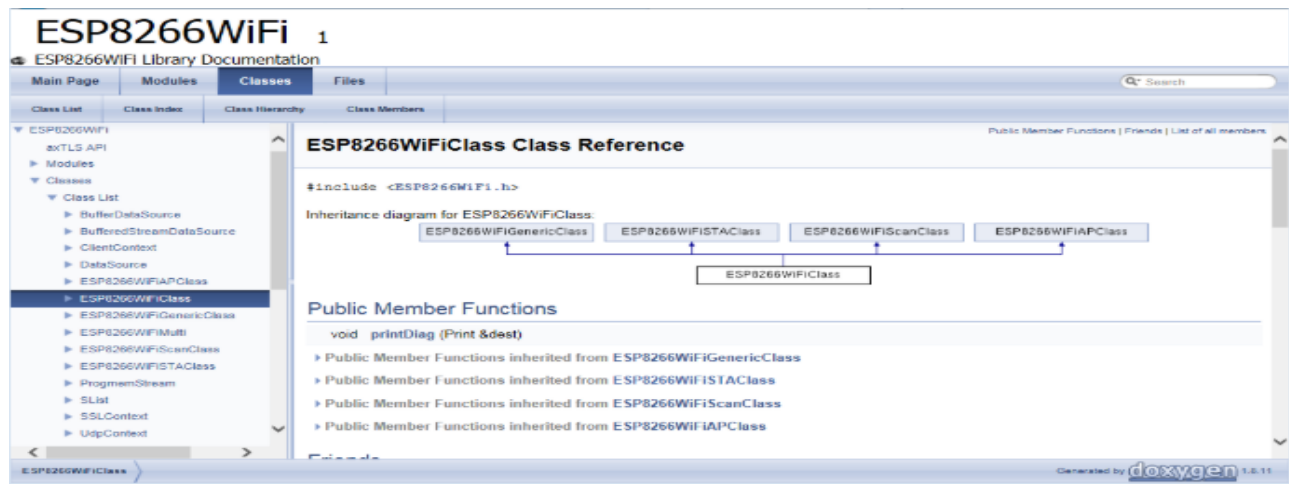


Рисунок 2.44. Документація ESP8266WIFI

Інструмент проскакує через усі заголовки та вихідні файли, збираючи інформацію з відформатованих блоків коментарів. Якщо розробник певного класу анотував код, ви побачите його, як у прикладах нижче.

```

wl_status_t ESP8266WiFiSTAClass::begin ( const char *   ssid,
                                           const char *   passphrase = NULL,
                                           int32_t        channel = 0,
                                           const uint8_t * bssid = NULL,
                                           bool           connect = true
                                           )

```

Start Wifi connection if passphrase is set the most secure supported mode will be automatically selected

Parameters

ssid const char* Pointer to the SSID string.
passphrase const char * Optional. Passphrase. Valid characters in a passphrase must be between ASCII 32-126 (decimal).
bssid uint8_t[6] Optional. BSSID / MAC of AP
channel Optional. Channel of AP
connect Optional. call connect

Returns

Definition at line 97 of file ESP8266WiFiSTA.cpp.

Рисунок 2.45. Перший приклад анотації

```

bool ESP8266WiFiSTAClass::hostname ( char * aHostname )

```

Set ESP8266 station DHCP hostname

Parameters

aHostname max length: 32

Returns

ok

Definition at line 422 of file ESP8266WiFiSTA.cpp.

Рисунок 2.46. Другий приклад анотації

Бібліотека SoftwareSerial

Апаратне забезпечення Arduino має вбудовану підтримку послідовного зв'язку на штирях 0 і 1 (яка також переходить на комп'ютер через USB-з'єднання). Вбудована серіальна підтримка відбувається за допомогою апаратного забезпечення (вбудованого в чіп) під назвою UART. Це обладнання дозволяє мікросхемі Atmega отримувати послідовне спілкування навіть під час роботи над іншими завданнями, якщо в 64-байтовому послідовному буфері є місце.

Бібліотека SoftwareSerial була розроблена для забезпечення послідовного зв'язку на інших цифрових штирях Arduino, використовуючи програмне забезпечення для копіювання функціональності (звідси назва "SoftwareSerial").

Можливе наявність декількох послідовних портів програмного забезпечення зі швидкістю до 115200 bps. Параметр дозволяє перевернутий сигнал для пристроїв, які потребують цього протоколу.

Iarduino_DHT

Визначте номер контакту, до якого підключений наш датчик, і створіть DHT-об'єкт. У розділі налаштування нам потрібно ініціювати послідовне повідомлення, оскільки ми будемо використовувати послідовний монітор для друку результатів. За допомогою функції `read22()` ми будемо зчитувати дані з датчика і вводити значення температури та вологості у змінні `t` та `h`. Якщо ви використовуєте датчик DHT11, вам знадобиться функція `read11()`. Наприкінці ми надрукуємо значення температури та вологості на моніторі.

```
/* DHT11/ DHT22 Sensor Temperature and Humidity Tutorial
 * Program made by Dejan Nedelkovski,
 * www.HowToMechatronics.com
 */
/*
 * You can find the DHT Library from Arduino official website
 * https://playground.arduino.cc/Main/DHTLib
 */

#include <dht.h>

#define dataPin 8 // Defines pin number to which the sensor is connected
dht DHT; // Creates a DHT object

void setup() {
  Serial.begin(9600);
}

void loop() {
  int readData = DHT.read22(dataPin); // Reads the data from the sensor
  float t = DHT.temperature; // Gets the values of the temperature
  float h = DHT.humidity; // Gets the values of the humidity

  // Printing the results on the serial monitor
  Serial.print("Temperature = ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("    Humidity = ");
  Serial.print(h);
  Serial.println(" % ");

  delay(2000); // Delays 2 seconds, as the DHT22 sampling rate is 0.5Hz
}
```

Рисунок 2.47. Значення температури та вологості на моніторі

Після того, як ми завантажимо цей код на плату Arduino, результати датчиків температури та вологості з датчика можна побачити на послідовному моніторі. Ось вихідний код цього прикладу:

```

/* DHT11/ DHT22 Sensor Temperature and Humidity Tutorial
 * Program made by Dejan Nedelkovski,
 * www.HowToMechatronics.com
 */
/*
 * You can find the DHT Library from Arduino official website
 * https://playground.arduino.cc/Main/DHTLib
 */

#include <LiquidCrystal.h> // includes the LiquidCrystal Library
#include <dht.h>

#define dataPin 8
LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)
dht DHT;

void setup() {
  lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display
}

void loop() {
  int readData = DHT.read22(dataPin);
  float t = DHT.temperature;
  float h = DHT.humidity;
  lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will be displayed
  lcd.print("Temp.: "); // Prints string "Temp." on the LCD
  lcd.print(t); // Prints the temperature value from the sensor
  lcd.print(" C");
  lcd.setCursor(0,1);
  lcd.print("Humi.: ");
  lcd.print(h);
  lcd.print(" %");
  delay(2000);
}

```

Рисунок 2.48. Результати датчиків температури та вологості

Висновки до розділу

У данному розділі розглянуто компонентну базу Arduino UNO, датчики освітлення, температури, вологості, електромагнітний соленоїд тощо. Вибрані необхідні датчики, які оптимально визначають необхідні показники клімату. Вибрані інструменти для розробки WEB додатку. Розглянуті бібліотеки, необхідні для розробки програмного коду до Arduino. Наведено приклади роботи з Wi-Fi модулем ESP. Наведено приклади та схеми підключення для кожного датчика.

РОЗДІЛ 3. ПРОГРАМУВАННЯ ТА УСТАНОВКА ПІДСИСТЕМИ КЛІМАТКОНТРОЛЮ НА БАЗІ ARDUINO UNO

3.1. Реалізація та підключення до мікроконтролера

Фінальна версія схеми, має наступний вигляд, та розміщення датчиків:

1. Коридор – сенсор температури, реле та кінцевий вимикач;
2. Ванна кімната – датчик вологості, електромагнітний соленоїд, датчик MOSFET;
3. Спальня – фоторезистор, реле, мотор для відкриття або закриття жалюзі.

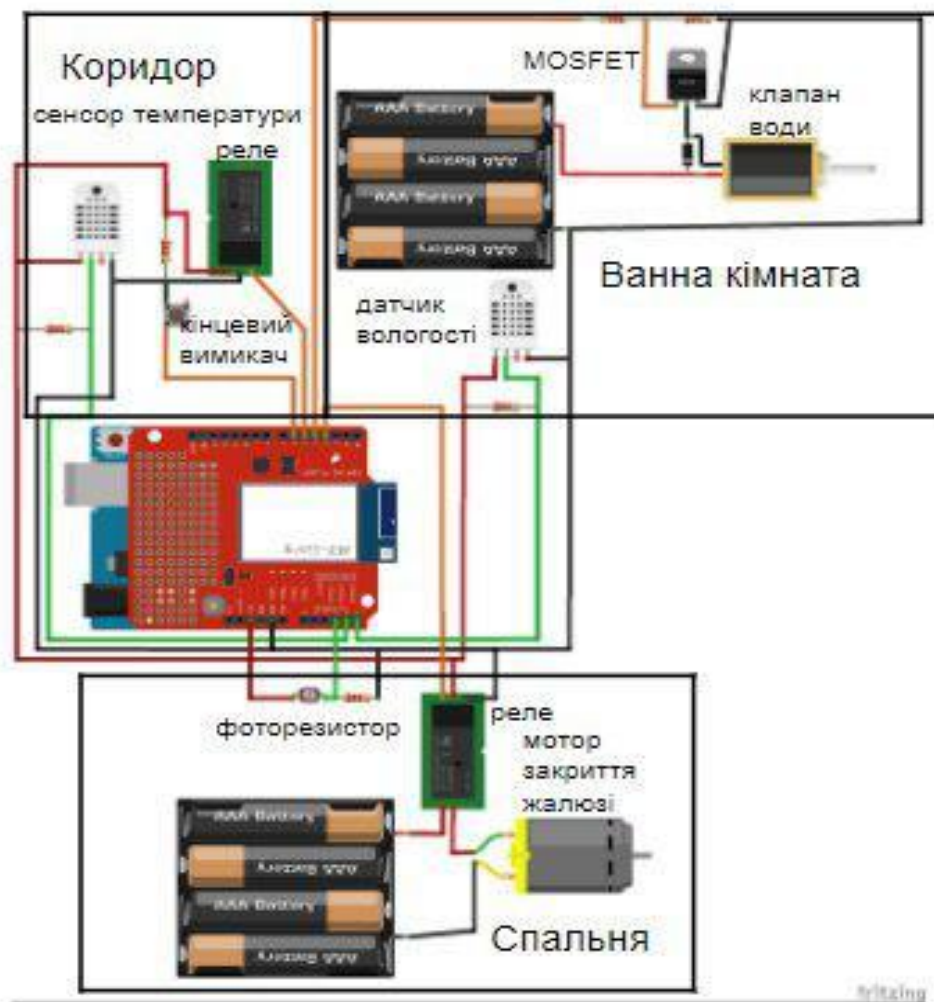


Рисунок 3.1. Фінальна схема підсистеми кліматконтролю

3.2. WEB додаток

Після з'єднання усіх вищеописаних датчиків, та підключення їх до мікропроцесора Arduino Uno, ми можемо приступити до оформлення WEB додатку.

Першим кроком є підключення вищеописаних бібліотек, таких як:

1. SoftwareSerial.h;
2. Iarduino_DHT;
3. ESP8266WIFI.

Наступним кроком, є визначення розташування датчиків, та їх ідентифікація у програмному коді. Потім для кожного з них потрібно створити змінну.

```
//-----підключення бібліотек-----
#include <SoftwareSerial.h>
SoftwareSerial ESPserial(2, 3); // RX | TX
#include <iarduino_DHT.h>
//|-----визначення контактів-----
iarduino_DHT sensor(6);
#define lock 8
#define conLim 35
#define humLim 90
#define conRelay 5
#define motorRelay 4
#define fotoResistor 0
#define tempSensor 5
#define lightLim 850
#define valvePin 7
//-----створення змінних-----
bool motor =false;
bool con = true;
bool conEsp = false;
bool doorStatus;
int light = 0;
int tempC;
unsigned long timing;
String sendToEsp;
String fullRequest[5];
```

Рисунок 3.2. Ініціалізація бібліотек та створення змінних

За допомогою функції `analog_Read()` відбувається считування інформації з датчиків температури та вологості. Але, для коректного відображення, температуру позначимо у Цельсіях, за допомогою наступної формули:

$$tempC = (tempC/1023.0)*5.0*100;$$

Наступним кроком був створений цикл, для датчика закриття та відкриття дверей, та датчики освітлення, що знаходяться за шторами показаний на рис. 3.3.:

```
tempC = analogRead(tempSensor); //зчитування інформації з датчика температури
tempC = ( tempC/1023.0 ) * 5.0 * 100; //формула для конвертування інформації з датчику температури у температуру в цельсіях
light = analogRead(fotoResistor); //зчитування інформації з датчику освітленості
sensor.read(); //зчитування інформації з датчику температури/вологості DHT11
doorStatus = digitalRead(lock); //зчитування інформації з кінцевого вимикача в дверях
//-----
if (doorStatus == HIGH){
    fullRequest[0] = '1'; //Якщо двері закриті, записуємо у нульовий біт передачі інформації до WiFi модуля ESP8266 - 1
}
else if (doorStatus == LOW){
    fullRequest[0] = '0'; //Якщо двері відкриті, записуємо у нульовий біт передачі інформації до WiFi модуля ESP8266 - 0
}
if (light < lightLim){
    fullRequest[4] = '1'; //Якщо дані з датчика освітленості менші за ліміт, зараз ніч, записуємо у четвертий біт передачі інформації до WiFi модуля ESP8266 - 1
}
else if (light > lightLim){
    fullRequest[4] = '0'; //Якщо дані з датчика освітленості більші за ліміт, зараз день, записуємо у четвертий біт передачі інформації до WiFi модуля ESP8266 - 0
}
temporaryString = tempC;
fullRequest[2] = tempC; //Записуємо у другий біт температуру
Serial.println(tempC);
temporaryInt = sensor.hum;
fullRequest[3] = temporaryInt; //Записуємо у третій біт вологість
```

Рисунок 3.3. Зчитування інформації з датчиків

Короткий цикл, що нарахує ступінь вологості:

```
if (sensor.hum > humLim){
    digitalWrite(valvePin, HIGH);
    fullRequest[1] = '1'; //Якщо вологість вище норми, перекриваємо воду та записуємо у перший біт 1
}
else {
    digitalWrite(valvePin, LOW);
    fullRequest[1] = '0'; //Якщо вологість нижче норми, записуємо у перший біт 0
}
sendToEsp = "";
sendToEsp = fullRequest[0] + fullRequest[1] + fullRequest[2] + fullRequest[3] + fullRequest[4] + fullRequest[5]; //Створюємо строку для передачі інформації до WiFi модуля ESP8266
char tempArray[7];
sendToEsp.toCharArray(tempArray, 7); //перетворюємо строку в масив типу char
ESPserial.write(tempArray); //Відправляємо масив інформації до WiFi модуля ESP8266
```

Рисунок 3.4. Цикл визначення ступеня вологості

Якщо температура в кімнаті оптимальна, та двері зачинені, можна виключити кондиціонер, заради економії електроенергії, а якщо навпаки – двері

зачинені, а температура зависока, або занижка, кондиціонер автоматично включається:

```
if (doorStatus == 1 && tempC > conLim && con){ //Якщо двері зачинені та температура в кімнаті вище норми, включаємо кондиціонер
    digitalWrite(conRelay, HIGH);
    con = false;
}
else if (con = false && doorStatus == 1 && tempC<conLim) { //Якщо двері зачинені, але температура в нормі, виключаємо кондиціонер
    digitalWrite(conRelay, LOW);
}

if (light < lightLim && motor == false){ //Якщо ніч, записати у змінну timing даний час та включити мотор закривання жалюзі
    motor = true;
    timing = millis();
    digitalWrite(motorRelay, HIGH);
    //Serial.println("HIGH");
}

if (millis() - timing > 4000){ //Через 4с від часу в змінній timing виключаємо мотор
    digitalWrite(motorRelay, LOW);
}
```

Рисунок 3.5. Код для датчика дверей

Для з'єднання мікропроцесора з мережею Інтернет, створюємо локальний сервер, та підключаємо WI-FI модуль:

```
//-----додавання бібліотек-----
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
//-----записуємо у масиви типу char ім'я та пароль WiFi точки-----
const char* ssid = "iPhone (Belyash)";
const char* password = "729294847";
//-----Створюємо змінні-----
String doorStatus;
String valveStatus;
String temperature1;
String humidity1;
String temperature2;
String humidity2;
String dayStatus;
String checkCodeChar;
String checkCodeString;
//-----Визначення пінів-----
#define conPin 5
#define valvePin 4
//-----Створюємо WiFi сервер-----
WiFiServer server(80);
```

Рисунок 3.6. Додавання бібліотек та змінних

Підключаємо Arduino UNO до точки доступу:

```
//-----Процес підключення до WiFi точки-----
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
server.begin();
Serial.println("Server started");
Serial.print("Use this URL to connect: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");
```

Рисунок 3.7. Підключення мікроконтролера до точки доступу

Тепер, треба провести тестування на коректне підключення модуля ESP8266. Потім, тестуємо локальну мережу:

```
if (request.indexOf("/CON=OFF") != -1) { //Якщо був отриман запит від сервера на вимкнення кондиціонера, посилаємо на пін conPin високий логічний рівень
    digitalWrite(conPin, HIGH);
    value = HIGH;
}
if (request.indexOf("/CON=ON") != -1){ //Якщо був отриман запит від сервера на вимкнення кондиціонера, посилаємо на пін conPin низький логічний рівень
    digitalWrite(conPin, LOW);
    value = LOW;
}
if (request.indexOf("/VALVE=OFF") != -1) { //Якщо був отриман запит від сервера на перекриття води, посилаємо на пін valvePin високий логічний рівень
    digitalWrite(valvePin, HIGH);
    valueValve = LOW;
}
if (request.indexOf("/VALVE=ON") != -1){ //Якщо був отриман запит від сервера на відкриття води, посилаємо на пін valvePin низький логічний рівень
    digitalWrite(valvePin, LOW);
    valueValve = HIGH;
}
serialRequest(); //Викликаємо функцію для конвертації інформації з мікроконтролера
```

Рисунок 3.8. Тестування локальної мережі

Відображаємо інформацію, отриману з датчиків, про двері та соленоїд. Якщо двері закриті або відкриті, система відобразить цю інформацію у WEB додатку. Якщо датчик вологості на полу зафіксує зависокий рієнь води,

електромагнітний соленоїд одразу ж перекриє клапан гарячої та холодної води.

```
//-----Виведення інформації про стан дверей, пори доби, статус електромагнітного клапана-----
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head><meta name='viewport' content='width=device-width, initial-scale=1' /> <meta charset='utf-8'><title>ESD 8266</title><style>button{color:red;padding: 10px 27px;}</>");
client.println("<h1 style='text-align: center;font-family: Open sans;font-weight: 100;font-size: 20px;'>SmartHome</h1><div>");
if (doorStatus == "OPEN"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;'>The door is OPEN</div>");
}
else if (doorStatus == "CLOSED"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>The door is CLOSED</div>");
}
if (dayStatus == "DAY"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;'>Now is DAY</div>");
}
else if (dayStatus == "NIGHT"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>Now is NIGHT</div>");
}
if (valveStatus == "OPEN"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;'>The water flow is OPEN</div>");
}
else if (valveStatus == "CLOSED"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>The water flow is CLOSED</div>");
}
if (value == HIGH){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>Air conditioning is now OFF</div>");
}
else if (value == LOW){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;'>Air conditioning is now ON</div>");
}
else if (valveStatus == "CLOSED"){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>The water flow is CLOSED</div>");
}
if (value == HIGH){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #e1212;margin: 0 auto;'>Air conditioning is now OFF</div>");
}
else if (value == LOW){
  client.println("<div style='text-align: center;width: 98px;color:white ;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;'>Air conditioning is now ON</div>");
}
}
```

Рисунок 3.9. Код датчика вологості

Далі, оброблена інформація з датчиків виводиться на екран в WEB додаток, та розбивається на такі пункти як:

1. Температура у приміщенні;
2. Вологість;
3. Включення або виключення кондиціонера;
4. Вода;
5. Освітлення.

```
//-----Виведення на сайт інформації про температуру, вологість, та створення кнопок ввімкнення/вимкнення кондиціонеру, води-----
client.println("<p><center>House temperature: ");
client.print(temperature1);
client.println(temperature2);
client.println("&deg;C </center>");
client.println("<p><center>House humidity: ");
client.print(humidity1);
client.println(humidity2);
client.println("%</center>");
client.println("<br>");
client.println("<center> Control of air conditioning system: </center>");
client.println("<div style='text-align: center;margin: 5px 0px;'> <a href='\"/CON=ON\"'><button>ON</button></a>&nbsp;<a href='\"/CON=OFF\"'><button>OFF</button></a></div>");
client.println("<center> Control of water flow: </center>");
client.println("<div style='text-align: center;margin: 5px 0px;'> <a href='\"/VALVE=ON\"'><button>ON</button></a>&nbsp;<a href='\"/VALVE=OFF\"'><button>OFF</button></a></div>");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
```

Рисунок 3.10. Вивід показників датчиків на екран

Останнім кроком є створення функції для конвертації інформації з мікроконтролера у змінні, для виведення її на сервер:

```
void serialRequest() { //-----Створення функції для конвертації інформації з мікроконтролера у змінні, для виведення на сервер-----
    char fullRequest[8];
    String string;
    bool check = false;
    if ( Serial.available() ){
        string = Serial.readString();
        string.toCharArray(fullRequest, 8);
        check = true;
    }
    checkCodeString = string;

    if (check){
        for (int i = 0; i < 8; i++){
            checkCodeChar += fullRequest[i];
        }
        if (fullRequest[0] == '0'){
            doorStatus = "OPEN";
        }
        else if (fullRequest[0] == '1'){
            doorStatus = "CLOSED";
        }
    }

    temperature1 = fullRequest[2];
    temperature2 = fullRequest[3];
}
```

Рисунок 3.11. Створення функції для конвертації інформації Цикл запитів, для датчика освітлення (день або ніч):

```
if (fullRequest[1] == '0'){
    valveStatus = "OPEN";
}
else if (fullRequest[1] == '1'){
    valveStatus = "CLOSED";
}

humidity1 = fullRequest[4];
humidity2 = fullRequest[5];

if (fullRequest[6] == '0'){
    dayStatus = "DAY";
}
else if (fullRequest[6] == '1'){
    dayStatus = "NIGHT";
}
check = false;
}
}
```

Рисунок 3.12. Цикл датчика освітлення

В результаті, був створений мультиплатформенний WEB додаток, в якому користувач може контролювати клімат у своєму домі.

На скріншотах 1, 2 та 3, можна побачити різноманітні варіації роботи програми з пристрою iPhone:

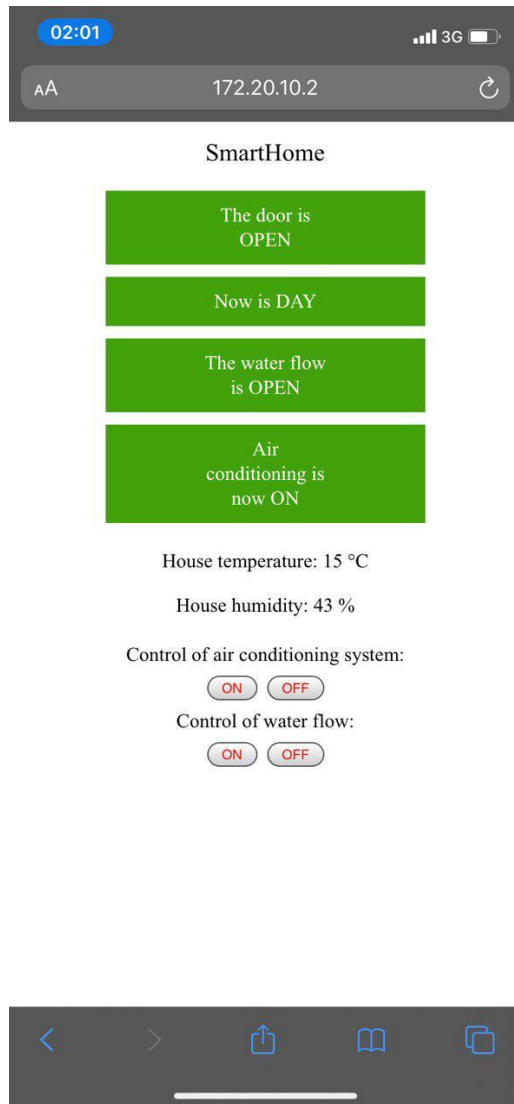


Рисунок 3.13. Скріншот 1

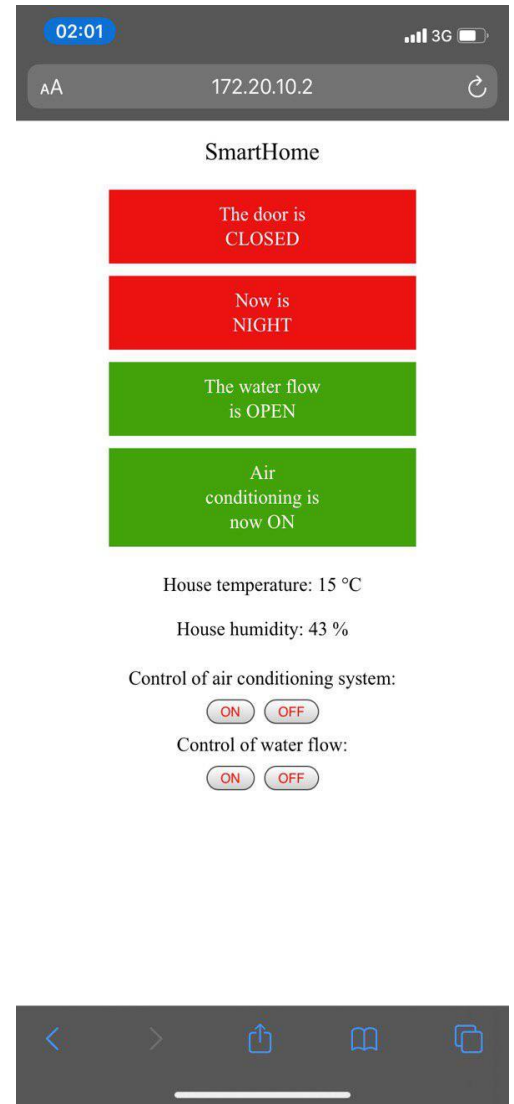


Рисунок 3.14. Скріншот 2

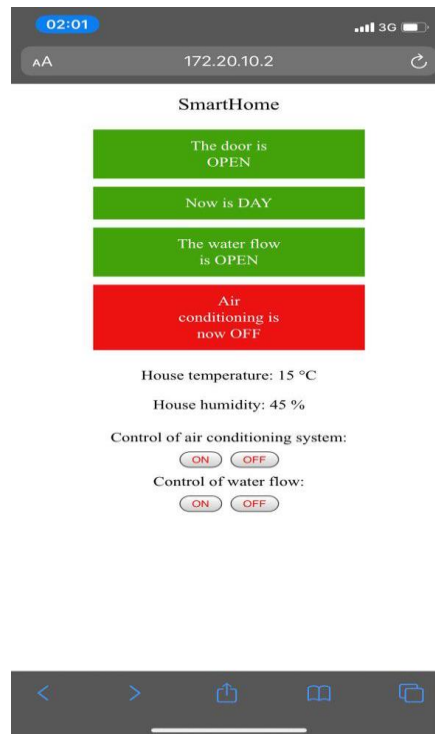


Рисунок 3.15. Скріншот 3

Висновки до розділу

Спроектована остаточна версія схеми підключення датчиків, спроектован WEB додаток, наданий опис ключових бібліотек та функцій програмного коду. Наведено схему підключення кожного датчику до мікроконтролера. Зроблена система для управління кліматом. Наведено приклади програмного коду для підсистеми кліматконтролю. Протестована остаточна версія моделі.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Узагальнено етапи розроблення стартап-проекту можна подати таким чином:

1. Маркетинговий аналіз стартап-проекту

В мережах цього етапу:

- 1) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- 2) аналізуються ринкові можливості щодо його реалізації;
- 3) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

2. Організація стартап-проекту

В межах цього етапу:

- 1) складається календарний план-графік реалізації стартап-проекту;
- 2) розраховується потреба в основних засобах та нематеріальних активах;
- 3) визначається плановий обсяг виробництва потенційного товару, на основі чого формується потреба у матеріальних ресурсах та персоналі;
- 4) розраховуються загальні початкові витрати на запуск проекту та планові загальногосподарські витрати, необхідні для реалізації проекту.

Фінансово-економічний аналіз та оцінка ризиків проекту В межах цього етапу:

- 5) визначається обсяг інвестиційних витрат;
- 6) розраховуються основні фінансово-економічні показники проекту

(обсяг виробництва продукції, собівартість виробництва, ціна реалізації, податкове навантаження та чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності проекту);

7) визначається рівень ризикованості проекту, визначаються основні ризики проекту та шляхи їх запобігання (реагування на ризики).

Заходи з комерціалізації проекту

Цей етап спрямовано на пошук інвесторів та просування інвестиційної пропозиції (оферти). Він передбачає:

- 1) визначення цільової групи інвесторів та опису їх ділових інтересів;
- 2) складання інвест-пропозиції (оферти): стислої характеристики проекту для попереднього ознайомлення інвестора із проектом;
- 3) планування заходів з просування оферти: визначення комунікаційних каналів та площадок та планування системи заходів з просування в межах обраних каналів;
- 4) планування ресурсів для реалізації заходів з просування оферти.

Означені етапи, реалізовані послідовно та вчасно – створюють передумови для успішного ринкового старту. Проте фахівці зі створення та розвитку стартаппроектів окремо відзначають, що відсутність маркетингових знань та умінь, що уможлиблюють розробку ринково затребуваного проекту із вихідної ідеї, є основною причиною високого рівня банкрутств стартапкомпаній, і ця проблема може бути вирішена за рахунок навчання винахідників. Відповідно, основним призначенням даних Методичних рекомендацій є надання студентам знань щодо суті, основних принципів розроблення стратегії ринкового впровадження та маркетингового управління інноваційними стартаппроектами у промислових галузях економіки, використання ефективних маркетингових інструментів просування високотехнологічних продуктів виробництва та послуг.

4.1. Опис ідеї проекту

В межах підпункту слід послідовно проаналізувати та подати у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- чим відрізняється від існуючих аналогів та замінників.

Зміст ідеї. Пропонується створити інтелектуальну систему медичної діагностики, яка основана на використанні штучних нейронних мереж і забезпечує підтримку прийняття рішень відносно діагнозу хвороби.

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів.

1. Конкуренти в Україні відсутні через практичну відсутність фахівців в галузі Розумного дому, слабка технічна забезпеченість державних медичних закладів, відсутність розуміння необхідності створення таких систем.

2. Потенційні конкуренти практично відсутні через відсутність кваліфікованих фахівців та бажанням заробляти великі та “швидкі” гроші.

3. Товарів замінників нема. Закордонні аналоги коштують занадто дорого, що робить їх економічно не вигідними в Україні.

4. Споживачі – звичайні люди, університети, школи.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Використання штучного інтелекту для обробки відео зображень УЗД та побудови підсистема підтримки прийняття рішень	1. Діагностика патологій	1. Діагностування різного виду патологій за допомогою апаратних та програмних засобів
	2. Допомога у встановленні клінічного діагнозу	1. Пришвидшення встановлення клінічного діагнозу, підвищення достовірності встановленого діагнозу

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабк а сторона)	N (нейтраль на сторона)	S (сильн а сторона)
		Мій проект	Nist			
1.	Кросплатформенність	Можливість використання на різних пристроях	На ПК під ОС Windows		+	
2.	Собівартість	Низька	Висока			+
3.	Зручність	Зручний інтерфейс	Інтуїтивно не зрозумілий		+	
4	Точність	Висока	Висока			+

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 4.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
 - чи існують такі технології, чи їх потрібно розробити/додати?
- чи доступні такі технології авторам проекту?

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Використання штучного інтелекту для обробки відео зображень УЗД та побудови підсистема підтримки прийняття рішень	Технологія 1 Використання нечіткого логічного висновку	наявна	доступна
		Технологія 2 Використання нечітких нейронних мереж	Необхідно доробити	доступна
		Технологія 3 Використання гібридних нейронних мереж	Необхідно доробити	доступна

Ідея проекту – створити робототехнічну підсистему кліматконтролю. Технологія 1. Використання мікропроцесора Arduino UNO. Дана технологія наявна. Авторам проекту вона доступна. Обрана технологія реалізації ідеї проекту – технологія 1.

Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Біля 30
2	Загальний обсяг продаж, грн/ум.од	100 млн.дол. США
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Дозвіл Міністерства охорони здоров'я
5	Специфічні вимоги до стандартизації та сертифікації	Відповідність національним стандартам
6	Середня норма рентабельності в галузі (або по ринку), %	R = 37%

На основі аналізу даних, наведених у таблиці можна зробити висновок, що ринок є привабливим для входження.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Високоєфективна медична діагностика захворювань щитовидної залози	медичні діагностичні заклади, лікарні, госпіталі, санаторії, медичні університети.	стандарти	Висока якість товару

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. № 4.6-4.7).

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Вартість діагностичного обладнання УЗД, КТ, МРТ	Підвищення ціни	Зміна постачальників
2	Висококваліфікований обслуговуючий персонал	Відсутність	Організація підготовки висококваліфікованого обслуговуючого персоналу

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Об'єм продажу	Ріст попиту	Поставки додаткової кількості систем
2.	Функціонал діагностичної системи	Розширення можливостей системи	Реалізація розширення можливостей системи

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 4.8)

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - олігополія	Домінує невелика кількість продавців, а вхід в галузь нових фірм обмежений високими бар'єрами.	Покращити рекламу
2. За рівнем конкурентної боротьби - національна	Між компаніями всередині країни	Розширити функціонал дії діагностичної системи
3. За галузевою ознакою - внутрішньогалузева	Між компаніями всередині країни	Знизити вартість програмного забезпечення
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між товарами одного виду	Покращити якість діагностування
5. За характером конкурентних переваг - цінова і нецінова	Ведуться цінові війни, а також в нецінові форматі, за рахунок кількості і якості послуг, що надаються	Покращити якість діагностування
6. За інтенсивністю - не марочна	Використовуються прийоми, що ставлять учасників ринку в нерівне становище	Знайти підтримку у Міністерстві охорони здоров'я

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера, додаток А) (табл. 4.9.).

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	- чи є можливості входу в ринок? Чи є потенційні конкуренти? Строки входу їх на ринок?	Чи постачальники диктують умови роботи на ринку. Які?	Чи клієнти диктують умови роботи на ринку. Які?	Обмеження для роботи на ринку через товари замінники

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентноспроможним на ринку.

На основі аналізу конкуренції, проведеного в п. 3.5 (табл. 4.9), а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 5) та факторів маркетингового середовища (табл. № 4.6-4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 4.10

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Алгоритмічне забезпечення	Використовується штучний інтелект
2	Програмне забезпечення	Використовуються сучасні пакети з штучними нейронними мережами

Робиться висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку.

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін
«Робототехнічна підсистема кліматконтролю»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства)						
			-3	-2	-1	0	1	2	3
1	Час складання	18				+			
2	Можливість аналізу результатів	15			+				
3	Комплексний підхід	10					+		
4	Надійність	15					+		
5	Простота реалізації	10			+				
6	Якість	17		+				+	
7	Технічне обслуговування	15						+	

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл.4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (про-гнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.12. SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1. Зменшення часу визначення діагнозу; 2. Зменшення фінансових затрат на дослідження; 3. Краща надійність; 4. Краща якість продукції; 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1. Складність у визначенні алгоритму; 2. Затрати на обладнання; 3. Унікальне ПЗ для обробки результатів;
<p>Можливості:</p> <ol style="list-style-type: none"> 1. Закріпитись на ринку товарів та послуг; 2. Конкурентоспроможність; 3. Іноземні інвестиції; 	<p>Загрози:</p> <ol style="list-style-type: none"> 1. Неможливість малих підприємств дозволити використання новітньої техніки;

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 4.9, аналіз потенційних конкурентів).

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Залучення нових споживачів — рекламувати товар в профільних установах для створення стійкого попиту та заохочення.	Ймовірність висока, оскільки якщо дана альтернатива виявиться новітньою розробкою, споживачі запитуватимуть продукт у посередників, а ті в свою чергу у виробника.	1 рік
2	Використовувати застарілий метод для роботи, адже більшість клієнтів на ринку не в змозі дозволити оновити виробництво	Ймовірність мала, оскільки існує альтернативна конкуренція на ринку.	6 місяців
3	Робота з сучасними представниками на ринку, в яких використання новітніх технологій витісняє застаріле.	Ймовірність висока, оскільки даний проект має велику кількість переваг.	5 місяців

Після аналізу зазначити обрану альтернативу. З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими.

Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Медичні діагностичні заклади	Готові	60%	Висока	Без обмежень
2	Лікарні	Готові	15	Середня інтенсивність	Без обмежень
3	Госпіталі	Готові	10	Низька інтенсивність	Без обмежень
4	Санаторії	Готові, при наявності потреби	5	Низька інтенсивність	Без обмежень
5	Медичні університети	Готові, при наявності потреби	10%	Низька інтенсивність	Без обмежень
Які цільові групи обрано: медичні діагностичні заклади, лікарні, госпіталі, санаторії, медичні університети.					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

- якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;
- якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;
- якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Встановлення низької ціни на новий товар для залучення більшої кількості покупців і завоювання великої долі ринку.	Стратегія диференційованого маркетингу	Комплексний підхід; оступність; ростота реалізації; Мобільність; изькі затрати.	Стратегія диференціації

Висновки: обрана стратегія розвитку спеціалізація через існування на ринку більш сильніших гравців. На перших кроках існування проекту доцільніше обрати стратегію спеціалізації та зайняти певну стабільну нішу на ринку.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Забирати існуючих у конкурентів.	Ні. Суть товару збільшити точність встановлення діагнозу, та зменшити час на його встановлення	Стратегія виклику лідера.

Висновки: оскільки проект не є першопрохідцем та має суттєві переваги по відношенню до свого прямого конкурента, можливо обрати стратегію виклику лідеру.

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (табл. 4.16) розробляється стратегія позиціонування (табл. 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартаппроекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Забезпечує високу точність діагностики; знижує витрати; більша надійність	Стратегія диференціації	Комплексний підхід; Точність; Економічність;	За показниками якості; За сферою застосування; За різновидом товару.

Висновки: На ряду із використання інноваційних методів підвищення точності, проект повинен викликати асоціації у клієнта у гнучкості налаштування та діагностики, можливості після продажно́ї підтримки та високої якості виготовлення.

4.3. Розробка ринкової стратегії

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18. Визначення ключових переваг концепції потенційного

товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точність	Забезпечує високу точність діагностики.	Точність забезпечується використанням інноваційної технології.
2	Економічність	зменшуються витрати на встановлення діагнозу	Зменшується час на постановку діагнозу

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Опис базової потреби споживача, яку задовольняє товар (згідно концепції), її основної функціональної вигоди		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Економічні	Нм	Вр
	2. Технічні	М	Тх
	3. Надійності	М	Тл
	4. Технологічні	М	Тх
	Якість: патент на корисну модель,		
	Пакування		
	Марка: НТУУ «КПІ ім. І. Сікорського» Медична діагностична система		
III. Товар із підкріпленням	До продажу: можливість аналізу та удосконалення системи.		
	Після продажу можливість створення нового ПЗ та модифікація існуючого		
За рахунок чого потенційний товар буде захищено від копіювання: захист ПЗ			

Висновки: основними засобами захисту від копіювання є патентування програмних та апаратних рішень, що використовуються у приладі. Окрім того, захист програмного коду повинен запобігти копіювання програми.

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проєкт буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проєкту), яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20).

Таблиця 4.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товаризамінники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	80000 \$	120000 \$.	10-15 0000000 \$.	100000/1500 00\$.

Висновки: обрано середню цінову категорію, оскільки занадто висока ціна відлякує споживача.

Таблиця 4.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Звична купівля з деяким змінами, або модифікована закупівля. Вона передбачає придбання децю змінених товарів (послуг), або зміну ціни на товар (послугу), або зміну кількості	Потачальник виконує такі основні функції: транспортування, складування, зберігання, доробка, просування до оптових і роздрібних торгових ланкам.	0 (без посередників)	Власна система збуту

Висновки: основними каналами збуту є підписка та продаж.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Орієнтація на регулярні поставки	Формальні/неформальні канали комунікацій	Комплексний підхід; Доступність; Мобільність;	Інформування споживачів; Стимулювання продажу; Пошук вигідних партнерів;	Даний продукт є інноваційним та унікальним

Висновки до розділу

Першим кроком було відбір та висвітлення самої ідеї проекту. Для цього приведено назву проекту та можливі зацікавлені сторони, котрі будуть потенційними споживачами продукту та які саме ризики можуть бути під час реалізації. Аналіз слабких та сильних сторін дають можливість автору проекту визначити аспекти, на яких слід зробити ставку. Перелік слабких, сильних та нейтральних характеристик ідеї дає можливість до уявлення конкурентноспроможності запропонованого рішення. Наступним кроком проводився технологічний аудит проекту. Під час аудиту автор отримує можливість до розуміння кращої технології виконання. Виявлено що більшість технологій вже існує, однак використання останніх не дасть можливості мати переваги над конкурентами. Зважаючи на стрімкий розвиток технологій, для швидкого отримання ресурсів та прибутків, слід обрати кооперацію, як форму ринкового впровадження проекту. Тобто, для того щоб встигнути за ринком, слід не розроблювати систему з нуля, оскільки це займе доволі багато часу, а запропонувати лідерам ринку використання запропонованих програмних рішень. Звісно, це є альтернативою, оскільки в такому випадку ні одна зі сторін не отримає повного прибутку. Після аналізу всіх аспектів ринку, подальша імплементація проекту можлива. Але є одна важлива умова. Слід провести доволі сильну рекламну компанію, та донести до споживачів необхідність використання системи саме з підвищеною завадостійкістю та чому саме обрати нашу систему.

ВИСНОВКИ

У даній магістрській дисертації проведено аналіз існуючих рішень, розкрито поняття Розумний дім. Проведена компонента класифікація та аналіз Розумного дому, його переваги та недоліки, обрана методологія та мікроконтролер Arduino UNO.

По поставленій задачі, був визначений перелік необхідних датчиків та компонентів, необхідних для реалізації роботи. Такими компонентами виступили модуль MOSFET irf520, датчик DHT11, реле, датчик температури lm35, датчик GL5516, WiFi модуль – esp8266-07, соленоїдний клапан, транзистор TIP120, мотор на 9В тощо.

Наведений приклад установки та початкової роботи з середовищем Arduino IDE. Обрані бібліотеки для розробки, та розроблений програмний код для WEB додатку. Протестовані та включені до мікроконтролера всі датчики, показана кінцева схема підсистеми кліматконтролю Розумного дому. Підключений локальний сервер також протестований, та показує стабільні результати. Створений WEB додаток для управління кліматом у домі.

Створена система протестована, та працює стабільно, також вона є хорошим конкурентом на ринку аналогічних пристроїв, поставлена задача виконана.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) S. Misra, M. Muthucumaru, H. Salman. "System model for the internet of things." Security challenges and approaches in internet of things." Springer, Cham, pp. 5-17, 2017.
- 2) J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future generation computer systems, vol. 29 (7), pp. 1645-1660, 2013.
- 3) S. Pirbhulal, H. E. Zhang, M. Alahi, H. Ghayvat, S. Mukhopadhyay, Y. T. Zhang, W. Wu. "A novel secure IoT-based smart home automation system using a wireless sensor network." Sensors, vol. 17(1), 69, 2017.
- 4) L. Chapman, J. B. Simon. "High-resolution monitoring of weather impacts on infrastructure networks using the Internet of Things." Bulletin of the American Meteorological Society, 2018.
- 5) L. Pocero, D. Amaxilatis, G. Mylonas, I. Chatzigiannakis, "Open source IoT meter devices for smart and energy-efficient school buildings." HardwareX, vol. 1, pp. 54-67, 2017.
- 6) R. Morello, C. De Capua, G. Fulco, S. C. Mukhopadhyay. "A Smart Power Meter to Monitor Energy Flow in Smart Grids: The Role of Advanced Sensing and IoT in the Electric Grid of the Future." IEEE Sensors Journal vol. 17.23, pp. 7828-7837, 2017.
- 7) A. Malhotra, S. Saini, V. V. Kale. "Automated Irrigation System with Weather Forecast Integration" , International Journal of Engineering Technology, Management and Applied Sciences, v. 5 (6), pp. 179- 184, 2017
- 8) D. H. Patel, D. Monali. "IOT Compatible Wireless Smart Portable Mini Weather Analyzer." International Research Journal of Engineering and Technology, vol. 4(5), 2017.
- 9) M. Taştan, "IoT Based Wearable Smart Health Monitoring System", Celal Bayar University Journal of Science, 14(3), pp. 343-350, 2018.

- 10) S. Ghanavati, H. J. Abawajy, D. Izadi, A. A. Alelaiwi. "Cloud-assisted IoT-based health status monitoring framework." *Cluster Computing*, vol. 20(2), pp. 1843-1853, 2017.
- 11) S. Deshkar, R. A. Thanseeh, G. M. Varun. "A Review on IoT based m-Health Systems for Diabetes." *International Journal of Computer Science and Telecommunications*, vol. 8(1), pp. 13-18, 2017.
- 12) M. Roopaei, R. Paul, R. C. Kim-Kwang. "Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging." *IEEE Cloud Computing*, vol. 4(1), pp. 10-15, 2017.
- 13) Gökozan, H., Taştan, M., Sarı, A., "Smart Cities and Management Strategies", II. International Strategic Research Congress, Antalya, Turkey, 2017, pp. 327-331
- 14) S. Samuel, I. Sujin. "A review of connectivity challenges in IoT-smart home." *Big Data and Smart City (ICBDSC)*, 2016 3rd MEC International Conference on. IEEE, 2016.
- 15) B. Alohalı, M. Merabti, K. Kifayat. "A secure scheme for a smart house based on Cloud of Things (CoT)." In *Computer Science and Electronic Engineering Conference (CEEC)*, 2014 6th , pp. 115-120, 2014.
- 16) B. H. Vishwajeet, W. Sanjeev "i-learning IoT: An intelligent self learning system for home automation using IoT." *Communications and Signal Processing (ICCSP)*, 2015 International Conference on. IEEE, 2015.
- 17) X. Ye, J. Huang. "A framework for cloud-based smart home. In *Computer science and network technology (ICCSNT)*, 2011 international conference on, IEEE, vol. 2, pp. 894-897, 2011.
- 18) https://en.wikipedia.org/wiki/Arduino_Uno
- 19) http://www.rusnauka.org/cgi-bin/search/step7_info.cgi?id=275800&idw=zROkaRdxm9OheWFPsa -

ДОДАТКИ

ДОДАТОК А СТАТТЯ: “РОБОТОТЕХНІЧНА СИСТЕМА “РОЗУМНОГО ДОМУ” НА БАЗІ ARDUINO”

Гайовий О.А.

*Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»*

РОБОТОТЕХНІЧНА СИСТЕМА “РОЗУМНОГО ДОМУ” НА БАЗІ ARDUINO

Вбудовані фізичні пристрої, такі як побутова техніка, стають розумнішими та розумнішими, вони оснащені вбудованими мікропроцесорами та бездротовими приймачами, що пропонують обмежені можливості зв'язку та забезпечують гарну поведінку. . Датчики та бездротові сенсорні мережі розгортаються в розумних домашніх рішеннях, точно вимірюючи умови навколишнього середовища в домашніх умовах, їх вдосконалені функції чутливості та підвищуючи їх точність, дозволяють розвивати розумні домашні додатки, які пропонують розширену автоматизацію. Автоматизація будинку включає в себе централізоване управління освітленням, кондиціонуванням, вентиляцією та опаленням, а також пристроями контролю та іншими. Автоматизація Smart Home, спрямована на підвищення ефективності та безпечності споживання енергії у вітчизняних сценаріях. Будинки оснащені автономними панелями управління для управління всіма системами та пристроями, присутніми в будинку. Більше того, ці панелі управління переважно не стосуються одне одного. Основна мета розумного дому - сфокусувати управління всіма пристроями в приватний блок управління, який може бути запрограмований на виконання спеціальних завдань, що підходять для власника та відповідного будинку. Метою розумного дому є не тільки придатність, але й зменшення споживання ресурсів, таких як електроенергія, газ тощо. Якщо у людини є можливість контролювати домашню автоматизацію, він може дистанційно зменшити споживання енергії і, таким чином, скоротити витрати. Екологічна стійкість набула актуальності в останні роки. Якщо нікого немає вдома, можна виключити кондиціонер та вентиляцію, для більш розумного споживання електроенергії. Це ж правило стосується освітлення, опалення та інших приладів. Деякі системи розумних будинків призупиняють роботу пристроїв, поки вони знову не знадобляться. Крім того, існує кілька різних технологій дослідження такого розумного будинку. Деякі стандарти використовують складні протоколи зв'язку та керування електропроводкою; інші покладаються на вбудовані сигнали у існуючий ланцюг живлення будинку. Частини покладаються на радіочастотні (РЧ) сигнали, а інші стають гібридами, поєднуючи кілька методів, усі контрольні завдання виконуються через мікропроцесор.

Система складається з двох частин: По-перше, частина Ethernet, яка відповідає за надання інструкцій з управління побутовими приладами та

отримання відповідей від датчиків. Друга частина - це мікроконтролерний блок Arduino Mega, який відповідає за управління пристроями, які з'єднуються з реле та датчиками. Мікроконтролер - це керівник системи, який контролює та керує всією інформацією в системі. На рисунку 1 показана блок-схема системи.

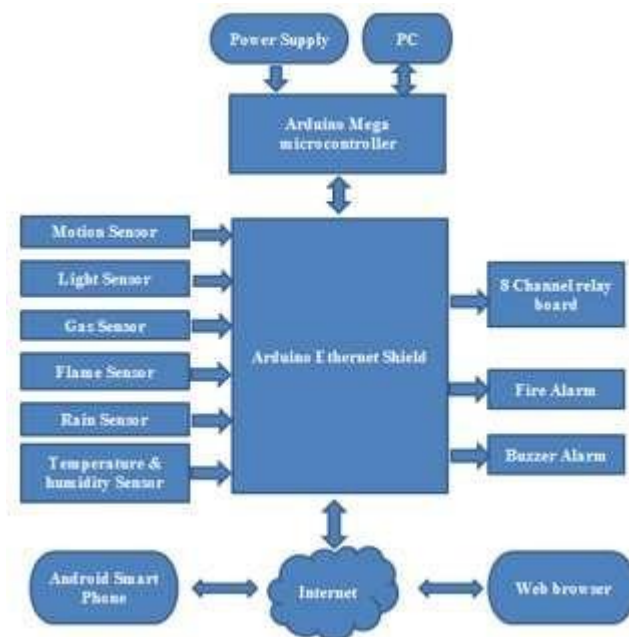


Рис.1 Блок-схема структури системи

У цій блок-схемі є дві основні важливі частини: плата мікроконтролера та модуль Ethernet. Частина модулю Ethernet, з'єднана з шістьма датчиками та восьми канальною платою реле, Датчик DHT11 - датчик температури та вологості в системі; датчик руху - детектор руху, який використовує пасивний інфрачервоний датчик; датчик дощу - це модуль YL-83; фоторезистор Arduino Electronics DIY Parts - це датчик світла; датчик полум'я є пожежним сповіщувачем, а датчик MQ5 - газовим детектором, також звуковим сигналом сигналізації тривоги та газовим зумом. Дані з усіх цих датчиків та реле постійно обробляються мікроконтролером, і статус надається у веб-браузерах та домашньому додатку, якщо в будинку є якась подія шляхом включення / вимкнення реле. Крім того, він може бачити або контролювати всі події та стан за допомогою веб-браузерів або домашньої програми.

Ця система була виконана на платформі Arduino Mega. Система використовує модуль Ethernet для взаємодії з Інтернетом для моніторингу, контролю приладів. Крім того, USB-порт використовується для з'єднання мікроконтролера з персональним комп'ютером для завантаження команди. На рисунку 2 показана остаточна робототехнічна система розумного дому.

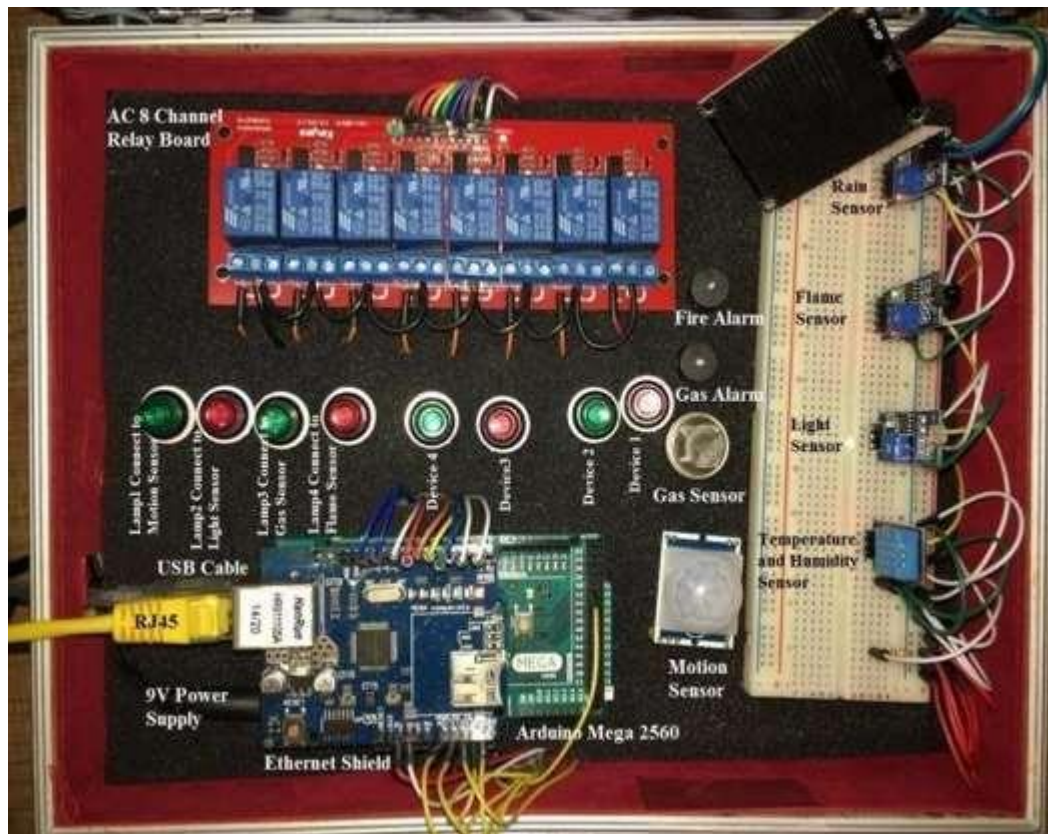


Рис. 2. Робототехнічна система розумного дому

Всі апаратні пристрої були перевірені та протестовані, щоб переконатися, що всі пристрої знаходяться в ідеальній робочій ситуації. Після цього всі пристрої були протестовані окремо за допомогою мікроконтролера, щоб отримати необхідне програмне забезпечення в цій реалізації та легко виявити будь-яку помилку на кожному апаратному пристрої. Тестом було встановлено, що всі апаратні пристрої працюють нормально, після цього потрібно було з'єднати всі апаратні пристрої між собою і протестувати загальну систему. Після тестування модуля Ethernet та його функціональності та підключення до мережі, система готова до тестування інших апаратних блоків, починаючи з сенсорних блоків: датчиків руху, світла, газу, полум'я та дощу; ці датчики є аналогами, тому їх потрібно перетворити з аналогового в цифровий формат. Блок-схема та схема підключення апаратних датчиків наведена на рисунках 3 та 4.

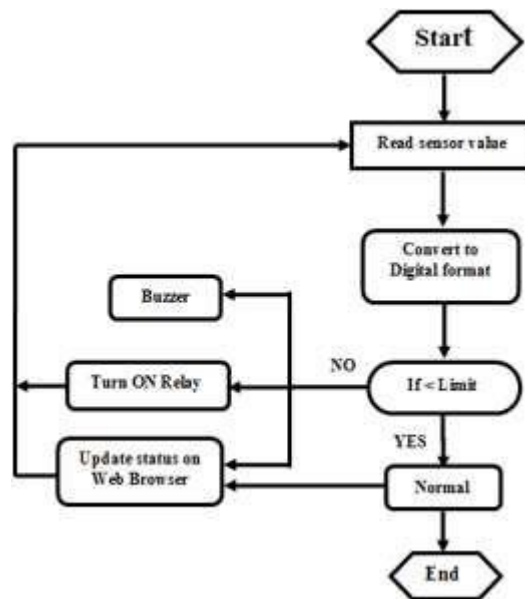


Рис. 3. Блок-схема для аналого-сенсорного програмного забезпечення



Рис. 4. Підключення апаратних датчиків

Програмне забезпечення веб-браузера персонального комп'ютера або будь-якого браузера смарт-пристроїв або програми Android використовується для апаратної системи для здійснення контролю за моніторингом системи через Інтернет. Це програмне забезпечення було створено в середовищі Arduino IDE. За допомогою цього програмного забезпечення власник може відслідковувати фактичне значення датчика, контролюючи електронні пристрої та бачити інформацію від мікроконтролера. Спочатку для входу на веб-сторінку програмному забезпеченню потрібні ім'я користувача та пароль на рисунку 5, показані вимоги автентифікації ім'я користувача та пароль для входу на головну сторінку.

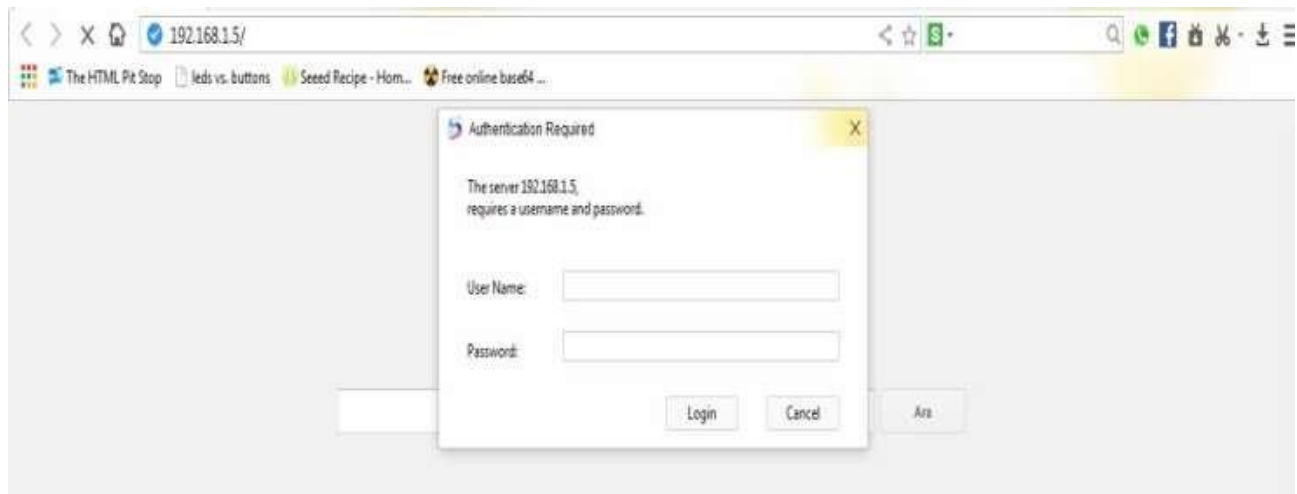


Рис. 5. Вимоги аутентифікації для входу

Веб-сторінка програми, що показує значення датчика та стан реле, показана на рисунку 6. Ця сторінка складається з трьох частин: перші вісім прямокутників представляють статус реле; друга чотири кнопки показують кнопку перемикавання з електронними пристроями ON / OFF, а третя частина показує поточні значення датчика читання. Більше того, система складається з шести датчиків; По-перше, датчик руху, який виявляє будь-який рух у будинку, увімкне реле вісімки, які підключаються до лампи навколишнього руху при перевищенні граничного значення; Друга частина, датчик світла, який використовується для автоматичного освітлення зовні будинку, увімкне реле сім, які підключаються до ламп у середовищі датчика світла при перевищенні граничного значення; Третя частина, датчик газу, який виявляє витoki газу в будинку, увімкне реле шість, які підключаються до вентилятора в середовищі витоку газу, також він активує зумер при перевищенні граничного значення; Четверта частина, датчик полум'я, що виявляє будь-яку пожежу в будинку, увімкне реле п'ять, які підключаються до пожежі у пожежному середовищі, також він активує пожежну сигналізацію при перевищенні граничного значення; По-п'яте, датчик дощу, який виявляє будь-який дощ зовні будинку, буде відображати стан дощу при перевищенні граничного значення; Заклучна частина, датчик температури та вологості, який відображає температуру та вологість всередині будинку.

Метою цього дослідження було розробити та виконати систему розумної автоматизації будинку, дистанційно керуючи та контролюючи електронні пристрої в будинку та моніторинг ступеня температури та відсотка вологості, витоку газу, пожеж та дощу, автоматизація світла за допомогою датчиків руху та світла всередині та зовні будинку, використовуючи Інтернет.



Рис. 6. Веб-тестування з показниками датчиків

Ці пристрої контролюються веб-браузером через Інтернет. Як результат, систему можна контролювати через веб-браузер. Крім того, датчики газу, полум'я, світла, температури та вологості та дощу працюють правильно з відмінними результатами. Ця система також використовує енергозбереження за допомогою датчика руху, світла для вмикання та вимикання освітлення в приміщенні та зовні. Важливою перевагою цієї системи є те, що контролювати кожен пристрій через веб-браузер та додаток Android в будь-якому місці та в будь-який час. Розроблена система пропонує гнучку, економічну та дистанційно керовану систему автоматизованого розумного будинку.

Література:

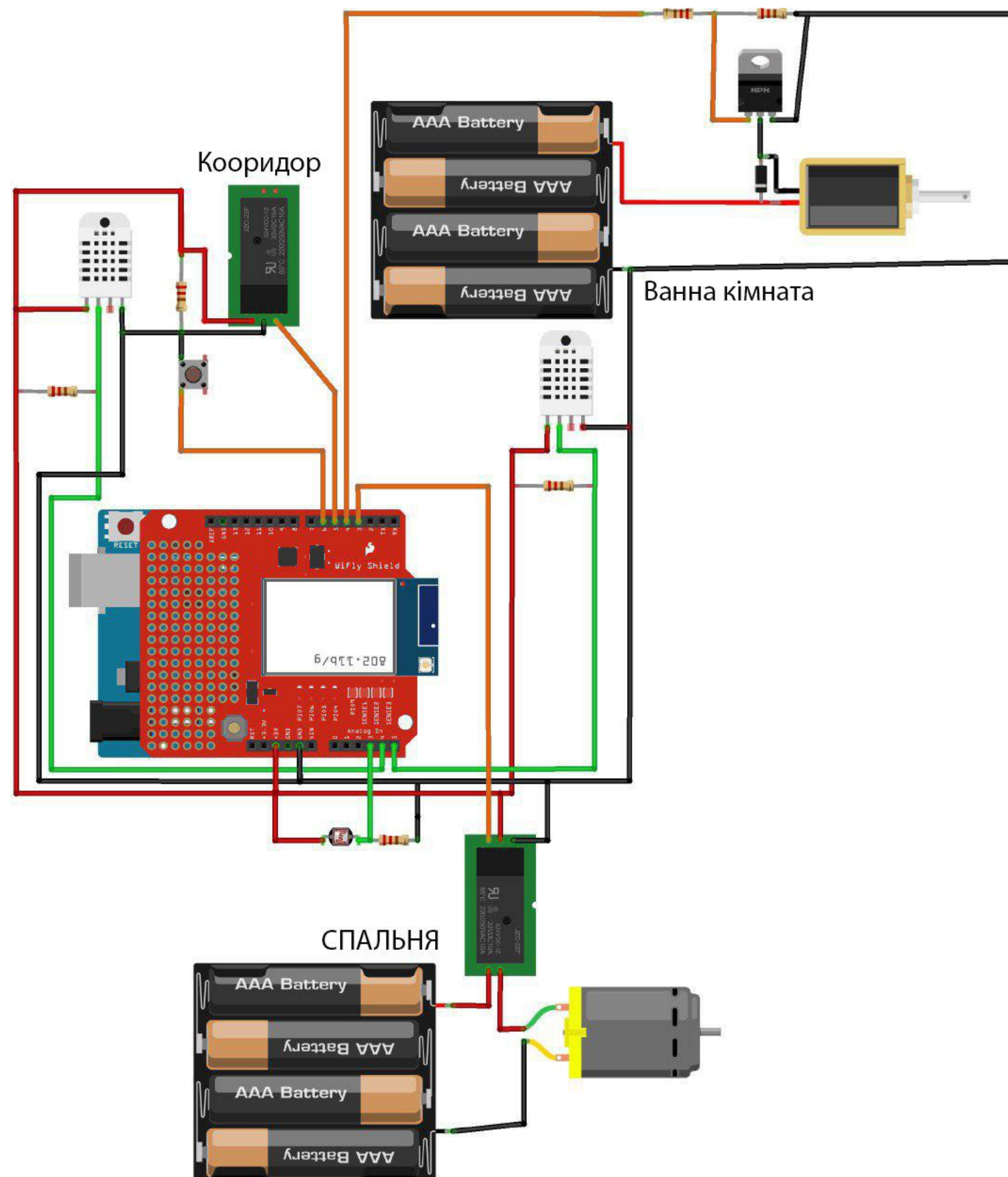
1. Kamilaris, A., "Enabling smart homes using web technologies," Ph.D. dissertation, Dept. of computer science, University of Cyprus, Cyprus, 2012.
2. Bergman E. Information appliances and beyond. Interaction design for consumer products. Morgan Kaufmann, pp.2-10, 2000.
3. Augusto, J.C., and Nugent, C.D. Smart homes can be smarter. Springer Berlin Heidelberg, pp. 5-12, 2006.
4. Staroch P., "A Weather Ontology for Predictive Control in Smart Homes," MSc dissertation, Dept. of Software Engineering and Internet Computing, Vienna, Austria, 2013
5. Algoiare T. O., "Design and Implementation of Intelligent Home Using GSM Network" MSc dissertation, Dept. of Computer Engineering, Ankara, Turkey, 2014.

ДОДАТОК Б: ПЕРЕВІРКА НА СПІВПАДІННЯ

ДОДАТОК В: ПЕРЛІК ІЛЮСТРАЦІЙНОГО МАТЕРІАЛУ

ПЛАКАТ 1: СХЕМА РОБОТЕХНІЧНОЇ ПІДСИСТЕМИ

СХЕМА РОБОТОТЕХНІЧНОЇ ПІДСИСТЕМИ

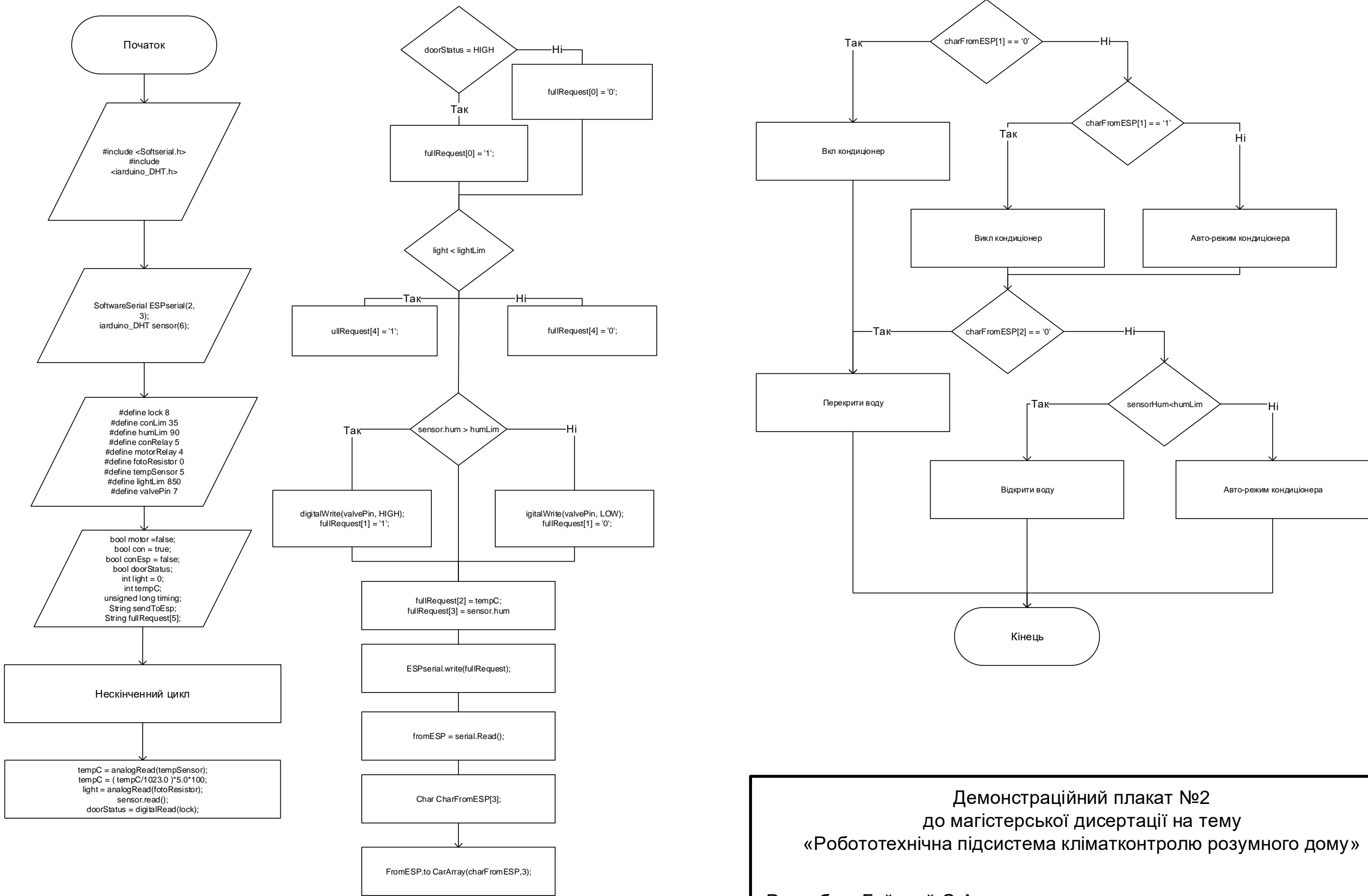


Демонстраційний плакат №1
до магістерської дисертації на тему
«Робототехнічна підсистема кліматконтролю розумного дому»

Розробив: Гайовий О.А.
Прийняв: д.т.н., ст.н.с., доцент Борукаєв З.Х.

ПЛАКАТ 2: СХЕМА УПРАВЛІННЯ КЛІМАТКОНТРОЛЕМ

СХЕМА УПРАВЛІННЯ КЛІМАТКОНТРОЛЕМ

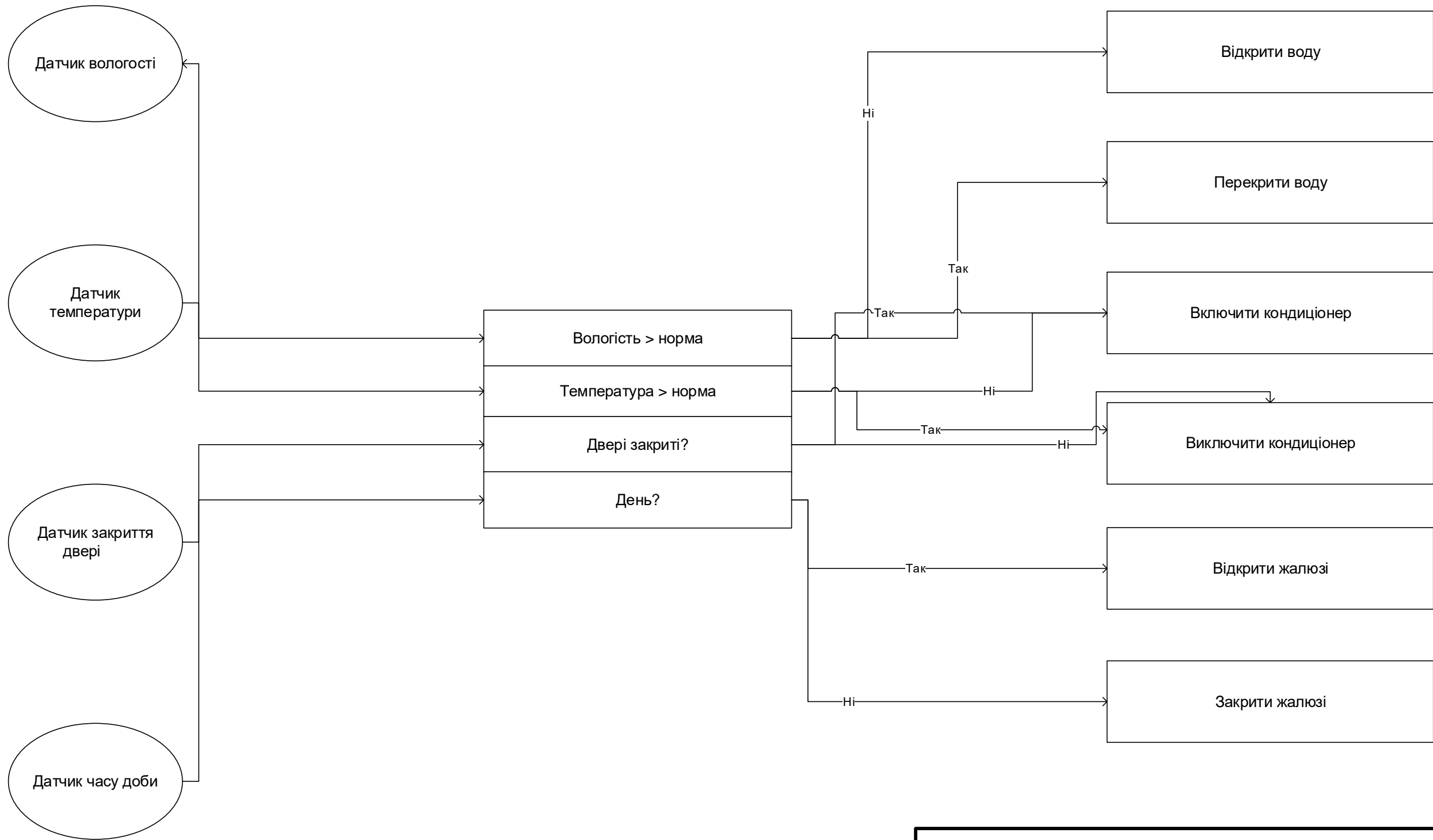


Демонстраційний плакат №2
до магістерської дисертації на тему
«Робототехнічна підсистема кліматконтролю розумного дому»

Розробив: Гайовий О.А.
Прийняв: д.т.н., ст.н.с., доцент Борукаєв З.Х.

ПЛАКАТ 3: СХЕМА ІНФОРМАЦІЙНИХ ПОТОКІВ

СХЕМА ІНФОРМАЦІЙНИХ ПОТОКІВ

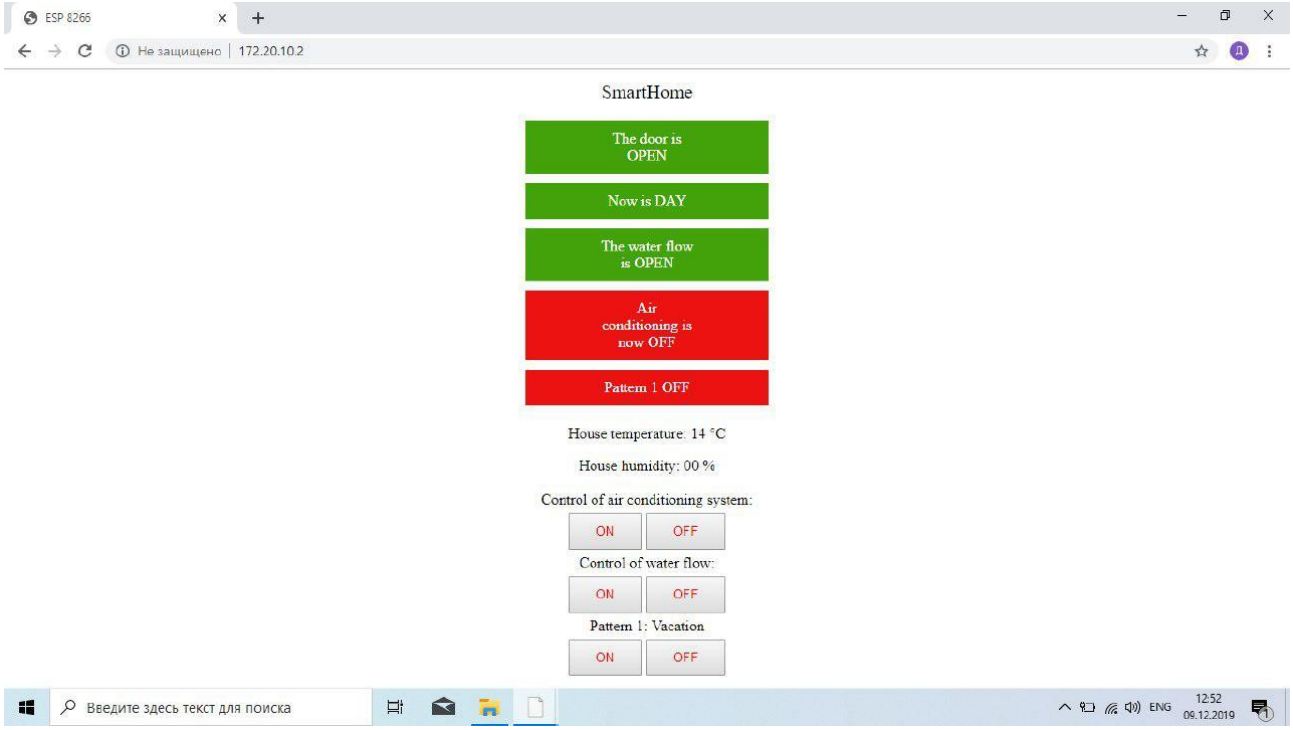


Демонстраційний плакат №3
до магістерської дисертації на тему
«Робототехнічна підсистема кліматконтролю розумного дому»

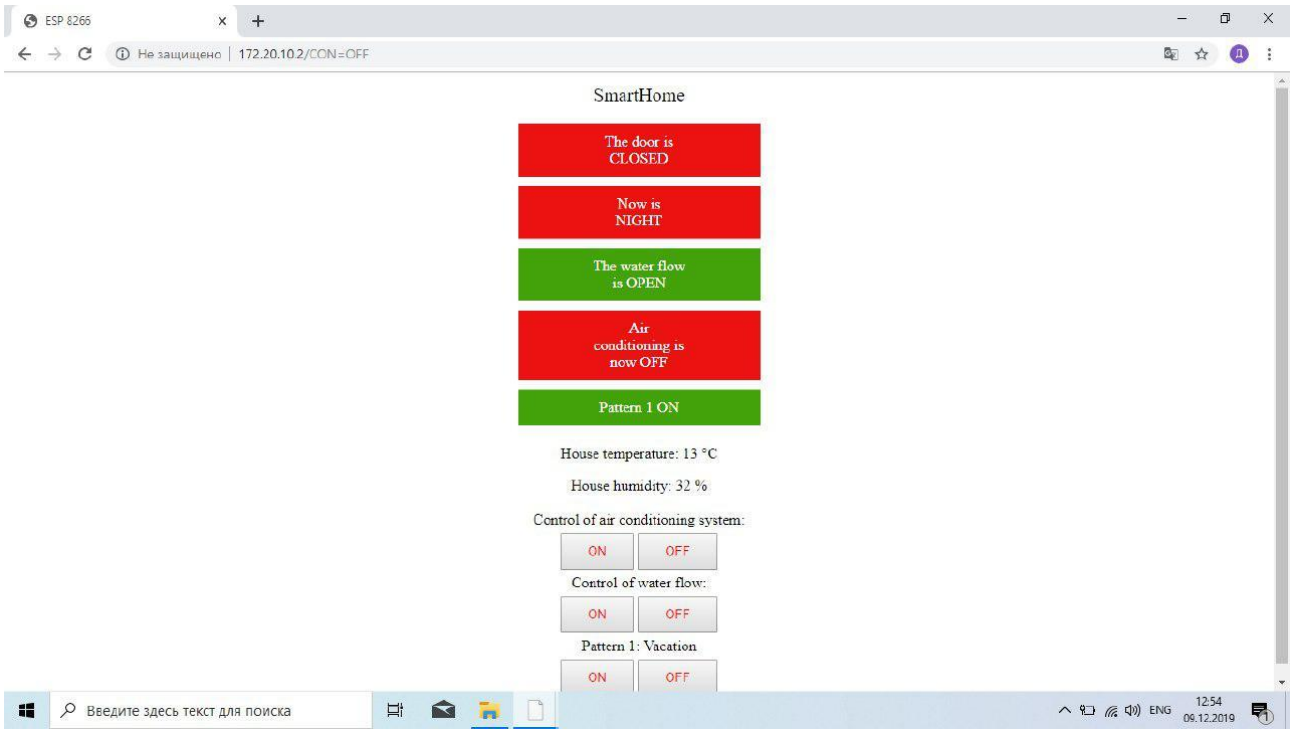
Розробив: Гайовий О.А.
Прийняв: д.т.н., ст.н.с., доцент Борукаєв З.Х.

ПЛАКАТ 4: СКРІНШОТИ РОБОТИ СИСТЕМИ

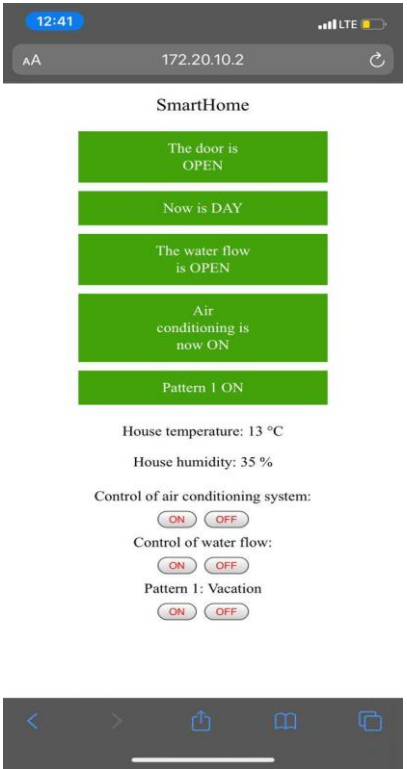
СКРІНШОТИ РОБОТИ СИСТЕМИ



Виключення патерна сценарія



Включення патерна сценарія



Вид з мобільного пристрою

Демонстраційний плакат №4
до магістерської дисертації на тему
«Робототехнічна підсистема кліматконтролю розумного дому»

Розробив: Гайовий О.А.
Прийняв: д.т.н., ст.н.с., доцент Борукаєв З.Х.

ДОДАТОК Г: ЛІСТИНГ

Лістинг:

```
//-----додавання бібліотек-----  
  
#include <ESP8266WiFi.h>  
  
#include <WiFiClient.h>  
  
#include <ESP8266WebServer.h>  
  
#include <ESP8266mDNS.h>  
  
//-----записуємо у масиви типу char ім'я та пароль WiFi точки-----  
-----  
  
const char* ssid = "iPhone (Belyash)";  
const char* password = "729294847";  
  
//-----Створюємо змінні-----  
  
String doorStatus;  
  
String valveStatus;  
  
String temperature1;  
  
String humidity1;  
  
String temperature2;  
  
String humidity2;  
  
String dayStatus;  
  
String checkCodeChar;  
  
String checkCodeString;  
  
//-----Визначення пінів-----  
  
#define conPin 5  
  
#define valvePin 4  
  
//-----Створюємо WiFi сервер-----  
  
WiFiServer server(80);  
  
void setup() {  
    pinMode(conPin, OUTPUT);  
    pinMode(valvePin, OUTPUT);  
    Serial.begin(9600);  
    delay(10);  
}
```

```

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

//-----Процес підключення до WiFi точки-----

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

server.begin();

Serial.println("Server started");

Serial.print("Use this URL to connect: ");

Serial.print("http://");

Serial.print(WiFi.localIP());

Serial.println("/");

}

void loop() {

    WiFiClient client = server.available();

    if (!client) {

        return;

    }

    Serial.println("new client");

    String request = client.readStringUntil('\r');

    Serial.println(request);

    client.flush();

    int value = HIGH;

```

```

int valueValve = HIGH;

if (request.indexOf("/CON=OFF") != -1) { //Якщо був отриман запит від сервера на
виключення кондиціонеру, посилаємо на пін conPin високий логічний рівень

    digitalWrite(conPin, HIGH);

    value = HIGH;

}

if (request.indexOf("/CON=ON") != -1){ //Якщо був отриман запит від сервера на включення
кондиціонеру, посилаємо на пін conPin низький логічний рівень

    digitalWrite(conPin, LOW);

    value = LOW;

}

if (request.indexOf("/VALVE=OFF") != -1) { //Якщо був отриман запит від сервера на
перекриття води, посилаємо на пін valvePin високий логічний рівень

    digitalWrite(valvePin, HIGH);

    valueValve = LOW;

}

if (request.indexOf("/VALVE=ON") != -1){ //Якщо був отриман запит від сервера на
відкриття води, посилаємо на пін valvePin низький логічний рівень

    digitalWrite(valvePin, LOW);

    valueValve = HIGH;

}

serialRequest(); //Викликаємо функцію для конвертації інформації з мікроконтроллера

//-----Виведення інформації про стан дверей,
пори доби, статус електромагнітного клапана-----

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("");

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\"/> <meta charset=\"utf-8\"><title>ESP 8266</title><style>button{color:red;padding:
10px 27px;}</style></head>");

```

```

client.println("<h1 style=\"text-align: center;font-family: Open sans;font-weight: 100;font-size:
20px;\">SmartHome</h1><div>");

if (doorStatus == "OPEN"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">The door is
OPEN</div>");

}

else if (doorStatus == "CLOSED"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">The door is
CLOSED</div>");

}

if (dayStatus == "DAY"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">Now is DAY</div>");

}

else if (dayStatus == "NIGHT"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">Now is NIGHT</div>");

}

if (valveStatus == "OPEN"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">The water flow is
OPEN</div>");

}

else if (valveStatus == "CLOSED"){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">The water flow is
CLOSED</div>");

}

if (value == HIGH){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">Air conditioning is now
OFF</div>");

}

```

```

else if (value == LOW){

    client.println("<div style=\"text-align: center;width: 98px;color:white ;border: 5px solid
#fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">Air conditioning is now
ON</div>");

}

//-----Виведення на сайт інформації про температуру, вологість, та
створення кнопок ввімкнення/вимкнення кондиціонера, води-----
---

client.println("<p><center>House temperature: ");

client.print(temperature1);

client.println(temperature2);

client.println("&deg;C </center>");

client.println("<p><center>House humidity: ");

client.print(humidity1);

client.println(humidity2);

client.println("%</center>");

client.println("<br>");

client.println("<center> Control of air conditioning system: </center>");

client.println("<div style=\"text-align: center;margin: 5px 0px;\"> <a
href=\"/CON=ON\"><button>ON</button></a>&nbsp;<a
href=\"/CON=OFF\"><button>OFF</button></a></div>");

client.println("<center> Control of water flow: </center>");

client.println("<div style=\"text-align: center;margin: 5px 0px;\"> <a
href=\"/VALVE=ON\"><button>ON</button></a>&nbsp;<a
href=\"/VALVE=OFF\"><button>OFF</button></a></div>");

client.println("</html>");

delay(1);

Serial.println("Client disconnected");

Serial.println("");

}

void serialRequest(){ //-----Створення функції для конвертації інформації з
мікроконтролера у змінні, для виведення на сервер-----

char fullRequest[8];

String string;

```



```
bool check = false;
if ( Serial.available() ){
    string = Serial.readString();
    string.toCharArray(fullRequest, 8);
    check = true;
}
checkCodeString = string;
if (check){
for (int i = 0; i < 8; i++){
    checkCodeChar += fullRequest[i];
}
if (fullRequest[0] == '0'){
    doorStatus = "OPEN";
}
else if (fullRequest[0] == '1'){
    doorStatus = "CLOSED";
}
temperature1 = fullRequest[2];
temperature2 = fullRequest[3];
if (fullRequest[1] == '0'){
    valveStatus = "OPEN";
}
else if (fullRequest[1] == '1'){
    valveStatus = "CLOSED";
}
humidity1 = fullRequest[4];
humidity2 = fullRequest[5];
if (fullRequest[6] == '0'){
    dayStatus = "DAY";
}
}
```

```

else if (fullRequest[6] == '1'){
    dayStatus = "NIGHT";
}
check = false;
}
}

//-----підключення бібліотек-----
#include <SoftwareSerial.h>
SoftwareSerial ESPserial(2, 3); // RX | TX
#include <iarduino_DHT.h>

//-----визначення контактів-----

iarduino_DHT sensor(6);

#define lock 8
#define conLim 35
#define humLim 90
#define conRelay 5
#define motorRelay 4
#define fotoResistor 0
#define tempSensor 5
#define lightLim 850
#define valvePin 7

//-----створення змінних-----

bool motor =false;
bool con = true;
bool conEsp = false;
bool doorStatus;
int light = 0;
int tempC;
unsigned long timing;
String sendToEsp;

```

```

String fullRequest[5];

void setup() {
  pinMode(lock, INPUT);
  pinMode(conRelay, OUTPUT);
  pinMode(conRelay, OUTPUT);
  pinMode(valvePin, OUTPUT);
  digitalWrite(conRelay, LOW);
  pinMode(motorRelay, OUTPUT);
  digitalWrite(motorRelay, LOW);
  ESPserial.begin(9600);
  Serial.begin(9600);
  delay(1000);
}

void loop() {
  char temporaryCharArray[2];
  String temporaryString;
  int temporaryInt;

  tempC = analogRead(tempSensor); //зчитування інформації з датчика температури
  tempC = ( tempC/1023.0 ) * 5.0 * 100; //формула для конвертування інформації з датчику
  температури у температуру в цельсіях

  light = analogRead(fotoResistor); //зчитування інформації з датчику освітленності
  sensor.read(); //зчитування інформації з датчику температури/вологості DHT11
  doorStatus = digitalRead(lock); //зчитування інформації з кінцевого вимикача в дверях
  //-----
  if (doorStatus == HIGH){
    fullRequest[0] = '1'; //Якщо двері закриті, записуємо у нульовий біт передачі інформації до
    WiFi модуля ESP8266 - 1
  }

  else if (doorStatus == LOW){
    fullRequest[0] = '0'; //Якщо двері відкриті, записуємо у нульовий біт передачі інформації
    до WiFi модуля ESP8266 - 0
  }
}

```

```

}

if (light < lightLim){

    fullRequest[4] = '1'; //Якщо дані з датчика освітленості менші за ліміт, зараз ніч,
    записуємо у четвертий біт передачі інформації до WiFi модуля ESP8266 - 1

}

else if (light > lightLim){

    fullRequest[4] = '0'; //Якщо дані з датчика освітленості більші за ліміт, зараз день,
    записуємо у четвертий біт передачі інформації до WiFi модуля ESP8266 - 0

}

temporaryString = tempC;

fullRequest[2] = tempC; //Записуємо у другий біт температуру

Serial.println(tempC);

temporaryInt = sensor.hum;

fullRequest[3] = temporaryInt; //Записуємо у третій біт вологість

if (sensor.hum > humLim){

    digitalWrite(valvePin, HIGH);

    fullRequest[1] = '1'; //Якщо вологість вище норми, перекриваємо воду та записуємо у
    перший біт 1

}

else {

    digitalWrite(valvePin, LOW);

    fullRequest[1] = '0'; //Якщо вологість нижче норми, записуємо у перший біт 0

}

sendToEsp = "";

sendToEsp = fullRequest[0] + fullRequest[1] + fullRequest[2] + fullRequest[3] + fullRequest[4] +
fullRequest[5]; //Створюємо строку для передачі інформації до WiFi модуля ESP8266

char tempArray[7];

sendToEsp.toCharArray(tempArray, 7); //перетворюємо строку в масив типу char

ESPserial.write(tempArray); //Відправляємо масив інформації до WiFi модуля ESP8266

if (doorStatus == 1 && tempC > conLim && con){ //Якщо двері зачинені та температура в
кімнаті вище норми, включаємо кондиціонер

    digitalWrite(conRelay, HIGH);

```

```
con = false;
}
else if (con = false && doorStatus == 1 && tempC < conLim) { //Якщо двері зачинені, але
температура в нормі, виключаємо кондиціонер
    digitalWrite(conRelay, LOW);
}
if (light < lightLim && motor == false){ //Якщо ніч, записати у змінну timing даний час та
включити мотор закривання жалюзі
    motor = true;
    timing = millis();
    digitalWrite(motorRelay, HIGH);
    //Serial.println("HIGH");
}
if (millis() - timing > 4000){ //Через 4с від часу в змінній timing виключаємо мотор
    digitalWrite(motorRelay, LOW);
}
//Serial.println();
delay(1000);
}
```